# Mixed Finite Element and Stochastic Galerkin Methods for Groundwater Flow Modelling: Efficiency Analysis and Real-Life Application

Luca Traverso

Dr Yuesuo Yang and Professor Tim Phillips

Submitted to
The School of Earth and Ocean Sciences and The School of Mathematics
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Cardiff University

April 2011

Thesis advisors

**Dr Yuesuo Yang and Professor Tim Phillips**

Author

**Luca Traverso**

# Abstract

Two research areas have received significant attention from the groundwater modelling community in recent years.

Firstly, the need for numerical techniques that are capable of generating accurate groundwater fluxes has been recognized in several groundwater related applications. The traditional approach is based on the solution of a second order problem which only provides an approximation for the potential. This is subsequently post-processed to derive an approximation of the groundwater fluxes. However, these approximations of the fluxes tend to be inaccurate. Mixed finite element methods (MFEM), based on the approximation of a first order problem, have emerged as a suitable alternative to the traditional approach since they provide accurate approximations for both the potential and the groundwater fluxes. However, the discrete linear systems obtained using mixed methods is indefinite and its solution is generally considered a source of problems. A variation of standard mixed methods enables the indefinite system to be transformed into a positive definite one for which standard iterative solvers can be used. In this thesis a comparison of the computational cost incurred in solving the indefinite and positive definite systems is presented. It is shown that the success of one method over the other is largely dependent on the choice of preconditioner used within the iterative scheme. Further evidence is provided which demonstrates that the Schur complement preconditioner proposed by Powell (2003) and Powell & Silvester (2003) for the indefinite system is robust and optimal for a class of conductivity coefficients often encountered in groundwater modelling applications.

Secondly, we provide an assessment of numerical methods for describing model uncertainty. This field of research has developed incredibly fast in the last decade with new advances being continuously proposed. In the context of groundwater modelling, uncertainty arises predominantly from scarce and erroneous knowledge of the hydraulic parameters of an aquifer. In a probabilistic framework these coefficients are modelled as spatial random fields and the *deterministic* partial differential equations (studied in the first part of the thesis) become *stochastic* in nature. In this thesis we study recently proposed methodologies to tackle uncertainty quantification. These belong to the large family of Stochastic Galerkin methods which use polynomial chaos expansions for the unknown solution. The conductivity coefficient is approximated by means of Karhunen-Loéve expansion (KLE) or by polynomial chaos expansion. The slow decay of the eigenvalues of the KLE for random fields with small correlation lengths poses a significant limitation to the applicability of this method since, in these circumstances, a large number of terms in the expansion (random variables) are required to attain reasonable accuracy. We show that this limitation can be overcome through a decomposition of the physical domain into regions whose sizes correspond approximately to the correlation lengths of the material parameters. This approach allows the deployment of expansions using a limited number of random variables. In this thesis we explore solution strategies for stochastic Galerkin methods. The characteristic structure of the discrete linear systems obtained when the underlying Galerkin method is either the Finite Element Method (FEM) or the mixed finite element method is described. The performance of iterative solvers preconditioned with traditional mean-based preconditioners is studied and it is shown that their performance deteriorates significantly for random fields characterized by large variances. For the stochastic primal formulation an alternative preconditioner based on a block symmetric Gauss-Seidel scheme is proposed and it is shown that it outperforms mean-based preconditioners for all settings considered in this work.

Our work concludes with the development of a numerical model for a real case study in the United Kingdom. A calibrated deterministic model for the site is developed using FEM and MFEM and then the calibrated model is used to obtain a probabilistic representation of the conductivity field. Thus stochastic technologies are deployed to quantify model uncertainty for the site. The reported case study is one of the first examples of formal characterization of model uncertainty for an actual site.

# Contents

# List of Tables

# List of Figures

# Acknowledgments

This research has been made possible by the PhD scholarship provided by Cardiff University School of Earth and Ocean Sciences and School of Mathematics.

I am wholeheartedly thankful to Professor Tim Phillips for his guidance and support throughout this project. I will always be grateful to him for taking on the challenge of supervising me and for giving me not only the knowledge and direction necessary for the completion of this project, but also a wider understanding and methodology for carrying out research. I am also thankful for his efforts to actively involve me in the activities of the Applied Mathematics research group which has given me the opportunity to validate the methodologies used in this research.

I would like to thank Dr Yuesuo Yang for his help with this project and also for giving me the opportunity to demonstrate and tutor on the courses of the MSc of Applied Hydrogeology.

The computational facilities and technical assistance provided by the Advanced Research Computing (ARCCA) team at Cardiff University has been essential for the successful completion of this project.

I would like to thank the Cardiff Harbour Authority for providing some necessary data for this project.

I would like to thank the Department of Mathematical Sciences at University of Bath for inviting me to give a talk at their Numerical Analysis seminar series. It has been one of the most rewarding experiences of my PhD journey and given me a great opportunity to publicise my research and gain confidence in my work.

Thanks to all the members of the Applied Mathematics research group for their constructive feedback about the project over the years.

I would like to sincerely thank Professor E.F Kaasschieter for his help with the code development of the Mixed Hybrid Finite Element method. I would also like to

thank Dr. Elisabeth Ullmann and Dr. Catherine Powell for kindly answering to my emails and for giving me useful suggestions and explanations.

Thanks to Ben for the great times spent in 2.31A, in Cardiff and elsewhere in Wales. I am also happy to have met Matt - a truly good person. The coffee clubs and poker nights have made my Cardiff experience quite memorable - thanks to Falko, Jenny, Tracy, Pedro, Martin, Brian, Alan and Denise for that. Thanks to Chris and all the other players of the Rolling Stones (the first and best team in the Earth Sciences) for the great Tuesday nights at Power League.

Grazie a Mariann, amore della mia vita. Senza il tuo supporto e incoraggiamento durante i tanti momenti difficili di questo Dottorato non so se ce l'avrei fatta a raggiungere la fine. Grazie per essere un punto di riferimento nella mia vita.

Grazie ai miei genitori. Se oggi ho raggiunto questo traguardo e' soprattutto merito vostro. La determinazione e consapevolezza nei miei mezzi che ho acquisito negli anni e' il frutto dei vostri insegnamenti. Spero che saro' per Sofia l'esempio che siete stati e siete tuttora per me.

And finally thanks to my parents for their constant support and for believing in me. Thanks to Pille and Ülo for caring about me and for the inspiring time spent in the summer house in the beautiful Estonian countryside.

To Mariann and little Sofia.

# Chapter 1

# Introduction

Groundwater models have, in recent decades, emerged as important tools used extensively by policy-makers and stakeholders in the sustainable management of water resources. Environmental regulators across the globe have adopted groundwater models to improve and support their decision-making. The United Kingdom (UK) has been at the forefront of these efforts. By 2006 the Environment Agency of England and Wales had sponsored the development of 34 catchment size groundwater models which altogether cover most of the major and some minor aquifers in the country (Van Wonderon & Wilson 2006).

Many processes in the physical sciences are mathematically described by partial differential equations (PDE). The movement of water in a porous medium is one of those processes. In fact, the combination of Darcy's Law and conservation of mass gives a second order partial differential equation. Provided that suitable boundary conditions are specified, its solution allows for the prediction of pressure and velocities everywhere in the physical domain under investigation. Generally, simple problems constituted by simple geometries and parameters admit analytical solutions. However, most often the modelling effort involves complex three-dimensional geometries and spatially varying parameter sets. In those circumstances analytical solutions are not available and one has to rely on numerical methods to obtain approximations for the quantities of interest.

Depending on the characteristics of the parameter datasets, boundary conditions and source / sink terms, numerical models can be either deterministic or stochastic. In the former case parameters, such as hydraulic conductivity are deemed to be known with certainty everywhere in the model domain. In the latter case it is recognized that such detailed knowledge is not available, thus the system parameters are described in probabilistic manner. In this thesis we investigate both modelling approaches. The deterministic approach is largely established and by far the most widely used in applications. Thus we deal with a specialised subfield - accurate approximation of groundwater fluxes by mixed finite element methods - in that area which has, however, significant relevance in specific groundwater modelling contexts. The stochastic approach has received extensive attention in the last decade due to the emergence of novel technologies. Therefore we investigate one of these technical advancements - stochastic Galerkin methods for uncertainty quantification - in a more holistic manner.

### Mixed and Hybrid Finite Elements: A Computational Comparison

Since numerical methods are not exact and they only provide an approximation of the actual solution, the research community has extensively focused on how to improve numerical model solutions. The accurate approximation of groundwater fluxes by sophisticated finite element methods represents an example of such achievements. Groundwater fluxes are often the variable of primary interest and their accurate evaluation is of crucial importance in many applications. As an example, the case of nuclear waste disposal can be considered. In this context groundwater velocities or fluxes are critical in determining the likely pathways and timings of radionuclides through the geological deposits, should they escape from the repository in which they are contained. Considering that the UK Government is aiming at nuclear energy as the primary source of its future energy supply and that geological disposal is the preferred option for dealing with nuclear waste, the research involved in predicting the fate of radionuclide dispersion in the underground is destined to increase significantly in the future.

Numerical methods that provide simultaneous accurate approximations of groundwater velocities and pressure head are available. Mixed Finite Element methods (MFEM) (Brezzi & Fortin 1991) were introduced in the early nineties and have been studied extensively in the last two decades. In the mixed formulation the coupled system of equations given by Darcy's Law and the conservation of mass is solved. This is different from the conventional approach where a single partial differential equation is solved for the pressure head and its post-processing gives the velocity solution. In the mixed approach, the velocity variable is defined explicitly by specific vectorial basis functions, thus no further post-processing is required. Importantly continuity conditions on the fluxes are imposed at the element level, thus making the method locally conservative and particularly suited for highly heterogeneous and discontinuous conductivity coefficients.

The fact that velocity approximations obtained using MFEM are superior to those obtained using traditional numerical methods has long been recognized and theoretical and discrete error estimates have mathematically proven it. However, it appears that the groundwater modelling community is generally unaware of locally conservative methods and instead software based on traditional numerical schemes are used for those applications (e.g. nuclear waste disposal) for which they are neither best suited nor recommended.

One of the reasons why MFEM have not gained the popularity that other methods have, is related to the fact that the associated discrete linear systems are indefinite and therefore generally more difficult to solve than symmetric positive definite (SPD) systems (generally obtained with traditional methods). Indefinite systems are considered problematic, making researchers to investigate ways to convert the indefinite systems to SPD ones. A popular approach is the hybridization method (Arnold & Brezzi 1985, Brezzi & Fortin 1991), also known as the Mixed Hybrid Finite Element Method (MHFEM).

Several authors (Younes & Fontaine 2008*b*,*a*) have compared the computational performance of various vectorial finite element schemes, but generally the hybrid version is considered in these studies. The original mixed method is discarded as the solution of a saddle-point system is considered, in principle, computationally too expensive and because the system of equations generated is larger than the one obtained with the hybrid method. However, there are several aspects that determine the computational cost of an iterative solver. The size of the system of equations is certainly one. Nevertheless, it would be superficial to discard one method based only on that criterion. In fact, considerations of the properties of the system of equations are equally important. The condition number, for example, gives an indication of the magnitude of change in the solution of a problem given small changes to model input parameters. Thus the condition number is influenced by several factors such as the size of the computational domain and more importantly the characteristics of the conductivity coefficient. If a system is ill-conditioned (large condition number), the iterative solver chosen for a specific problem is likely to perform poorly. In those cases its performance can be improved significantly, for example, by using a preconditioner. Therefore the number of unknowns is not a sufficient condition to determine if a method is more or less computationally expensive than another. In fact, a large system of equations can require a small number of solver iterations (and therefore computational cost) if an efficient preconditioner exists. Equally a small system of equations can require a large number of solver iterations if the condition number is large and an effective preconditioner is unavailable.

Following this discussion it is apparent that further investigation is required. Therefore, this thesis seeks to answer the following question for non-stochastic problems: under which circumstances is solving the indefinite system computationally more expensive than solving the positive definite system obtained with the hybrid approach?

In order to successfully answer this question, we consider iterative schemes equipped with state of the art preconditioners. The analysis includes test problems with various levels of mesh refinement, structured / unstructured meshes and heterogeneous, anisotropic and discontinuous conductivity coefficients. Each of the test problems considered possesses an analytical solution, and discrete error estimates are also included in the analysis.

The codes developed to carry out the numerical experiments associated with this analysis have all been developed within the MATLAB environment and the computations are all performed in serial. The development of the same algorithms in a parallel architecture is matter for future work and development.

### Stochastic Galerkin Methods for Uncertainty Quantification in Groundwater Flow Problems

The second part of this thesis is dedicated to the fascinating research area of uncertainty quantification (UQ). This topic has received significant attention in the last ten years as its relevance spans a variety of research areas of numerical analysis. The reviews by Najm (2009) and Le Maître & Knio (2010), for example, give an excellent overview of uncertainty quantification in computational fluid dynamics. Sudret & Der Kiureghian (2000) summarise the strengths and weaknesses of various methodologies with application to elasticity problems. An overview of innovative methods for uncertainty quantification in several areas of engineering and physical sciences is given by Stafanou (2009).

Deterministic models assume that coefficients, such as hydraulic conductivity or transmissivity and boundary conditions and source / sink terms are known with certainty in the physical domain. Unfortunately, this is never the case, for data used by numerical models are ordinarily uncertain. In fact, observed data are generally scarce and this leads to extrapolation to larger scales (often of the size of the computational domain) which is intrinsically uncertain. The lack of knowledge of the system parameters requires that uncertainties are quantified in a proper and satisfactory manner.

When the variables and coefficients of the groundwater flow equations are represented by random variables or random fields, the deterministic groundwater flow equations which are considered in the first part of this thesis become stochastic in nature. The efficient solution of stochastic PDE's (SPDE) poses a serious challenge as the number of equations which are solved are generally of several orders of magnitude larger than in deterministic problems. It becomes apparent that when uncertainty quantification is required for problems which are very large in nature (such as climate, ocean, reservoir or mantle models) the computational cost to carry out that task becomes prohibitively large.

Stochastic modelling of groundwater flow has been traditionally associated with *Monte Carlo* methods (MCM). This approach is straightforward for it involves the implementation of a large number of sequential deterministic simulations from which statistics of the numerical solutions can be derived. It is clear that the conclusions drawn in the first part of the thesis have immediate relevance to MCM, for their (computational) performance is directly proportional to the computational cost of solving the individual deterministic system.

However, traditional MCM are computationally expensive for a large number of simulations is generally required to compute meaningful statistics. This has led the research community to investigate alternative, faster converging methods or ways to accelerate the slow convergence of MCM. The latter research direction has resulted in the development of *Multilevel Monte Carlo* and *Quasi-Monte Carlo* methods which are giving very promising results (Cliffe et al. 2011, Graham et al. 2011). These methods are particularly suitable for those applications in which the stochastic behaviour requires a large number of degrees of freedom in probability space to be fully described. Situations of this kind are encountered in problems with rough coefficients (i.e spatial random fields with large variance and / or small correlation lengths). In such applications other methods such as *Stochastic Finite Element* method (SFEM) or *Stochastic Galerkin* method (SG) (Ghanem & Spanos 2003) and *Stochastic Collocation* method (SC) (Babuška et al. 2007) suffer from what is generally called 'curse of dimensionality' whereby the computational cost grows rapidly (factorially) with the dimension of the stochastic space.

Although this limitation of SFEM or SG is generally recognized, they continue to be widely used in engineering applications. In fact, in this thesis we aim to show that these methods can be successfully used in the context of groundwater modelling. Several studies have already reported work of various kinds in this specific area. However these are generally mathematical and somewhat technical. Often the examples used are 'toy' problems whose usefulness is restricted to the numerical analysis context. Therefore, we aim to apply these techniques to the Cardiff Bay case study and give one of the first examples of formal uncertainty quantification in a real-life situation. To achieve this, we assume that the highly heterogeneous conductivity field can be decomposed into sub-domains in which the material parameter has a quasi-homogeneous behaviour. This assumption which is perfectly justifiable and in line with approaches generally undertaken in applications, allows us to reduce the number of random variables required to approximate the conductivity field.

There are several ways material parameters can be described in a probabilistic manner. Generally, Gaussian, uniform or lognormal random variables are used for this scope. In the case of SFEM / SG methods the linear systems obtained from the variational formulation are significantly different depending on the distribution used to characterise the uncertain parameter/s. Thus if Gaussian and uniform distributions ('stochastically linear case') are employed the structure of the discrete system is considerably different from the case in which lognormal distributions are employed ('stochastically non-linear case'). In this thesis numerical analysis based on both cases is reported.

To be able to achieve our objective, which is the effective and efficient implementation of Stochastic Galerkin methods in groundwater modelling applications, there are several challenges which need to be overcome. First, given that the obtained discrete linear system of equations is of several orders of magnitude larger than its deterministic counterpart, the memory requirements for assembling such a large system pose serious challenges. However, as will be shown, this limitation can be overcome following the pioneering work of Ghanem & Kruger (1996).

Second, the system of equations has to be solved efficiently. We can build from our expertise with deterministic solvers investigated in the first part of the thesis. However, the stochastic Galerkin systems, in both stochastically linear and non-linear cases, are ill-conditioned with respect to the mesh size and the parameters defining the conductivity field. Thus, to effectively tackle the solution of such systems, preconditioners are required. A popular choice, which has been extensively exploited in the past, is the so called 'mean based preconditioner'. We assess its performance for a set of test problems, highlight the weaknesses and propose an alternative preconditioner for these challenging problems.

The code implementation of Stochastic Galerkin methods associated with the performance analysis presented in this thesis have all been developed within the MATLAB environment and in serial. The development of the same algorithms in a parallel architecture is matter for future work and development.

### Structure of the Thesis

The aim of this work is to analyse numerical methods for groundwater modelling, with special emphasis on finite elements, as these evolve from deterministic to stochastic formulation. Therefore the dissertation is structured around those two themes. Building on the extensive literature about classical FEM in the first part of the thesis we start our investigation with the mixed finite element method and report a comparison of computational performance between the classical MFEM and the hybrid approach. Considering the relative novelty of the stochastic approach, in the second part of the thesis we primarily focus on SFEM and advance subsequently to Stochastic Mixed Finite Element Method (SMFEM) which is currently an actively evolving field of research. A thorough comparison of solvers' performance is reported for both stochastic methods. The thesis concludes with an application of these methods to the Cardiff Bay case study, thus providing one of the first examples of the utilisation of stochastic technologies in a real-life scenario.

Following this general logic in *Chapter* 2 the theory of the mixed finite element method is presented. The derivation of the discrete linear system and the extension to the hybrid approach are described. Solution strategies for both methods are presented with particular emphasis on the state of art solvers currently available in the literature. *Chapter* 3 reports numerical experiments on the computational cost of solving the linear systems obtained by MFEM and MHFEM. The analysis is performed on structured / unstructured triangular and rectangular elements and for heterogeneous, anisotropic and discontinuous conductivity coefficients. MFEM discrete error estimates are reported for each test problem. *Chapter* 4 describes the theory of stochastic Galerkin methods for the stochastically linear case. The structure and properties of the the discrete linear systems for SFEM, SMFEM and the hybrid version of SMFEM are studied in depth. Existing solution strategies and innovative approaches are presented. The validation of SFEM and SMFEM against traditional MCM for a pair of test problems is given in *Chapter* 5. This chapter only serves as validation for SG methods and it is not intended to give a formal computational comparison between MCM and SG methods. Numerical experiments for the stochastically linear case are reported in *Chapter* 6. The first part of the chapter deals with SFEM and the second part with the SMFEM. Various solvers are tested and compared and the chapter ends with concluding remarks on which one is the most robust and computationally efficient. *Chapter* 7 follows the structure of the previous chapter, but considers the stochastically non-linear case. The first part of the chapter describes the derivation of the global linear system as this differs substantially from the linear case. *Chapter* 8 discusses the Cardiff Bay case study. The first part of the chapter outlines the conceptual model for the site and the second part shows the numerical simulations. Both deterministic finite element and mixed finite element simulations are included, as well as their stochastic counterparts. The thesis concludes with *Chapter* 9 which summarises the findings of this work, highlights the unanswered questions and outlines possible directions of future research.

# Chapter 2

# Mixed and Hybrid Finite Element Theory

## 2.1 Introduction

The importance of accurate approximation of fluxes in groundwater modelling has been at the heart of debates in this field for the last two decades. Accurate computation of the fluxes is important not only when the computed flow solution is used to solve the contaminant transport equations, but also when accurate water balances are required for the problem at hand. The finite element method (FEM), the finite difference method (FDM) and the finite volume method (FVM) are the most widely used numerical techniques for the approximation of groundwater fluxes. These numerical methods, first solve for the potential and then obtain the flux by numerical differentiation using Darcy's Law. A review of different Darcian post-processing methods is given by Goode (1990), Cordes & Kinzelbach (1992), Srivastava & Brusseau (1995), Dogrul & Kadir (2006). Whilst post-processing techniques might be suitable for problems with relatively homogeneous hydraulic conductivity, they are not appropriate for heterogeneous aquifers (Kaasschieter & Huijben 1992, Mosé et al. 1994). They are particularly prone to error when the hydraulic conductivity coefficient is discontinuous with large contrasts in different regions of the problem domain.

Mixed finite element methods (MFEM) (Arnold & Brezzi 1985, Brezzi & Fortin 1991) represent an alternative to traditional numerical schemes which allow the accurate simultaneous approximation of potential and groundwater fluxes. Mixed methods are based on the choice of vectorial basis functions as a suitable approximation space for the normal components of fluxes across each finite element edge or face. Additionally, scalar basis functions, which are element-wise constant, are chosen for the approximation of the potential. Mixed methods have the important advantages of being locally conservative and of enforcing continuity on the normal components of the fluxes at the finite element boundaries.

Groundwater fluxes obtained by mixed methods are generally more accurate than those obtained through Darcian post-processing and this has been demonstrated by several authors (see Durlofsky (1994), Kaasschieter (1995) for example). This is achieved at the expense of larger computational cost, simply because the number of degrees of freedom in the mixed formulation is larger than traditional methods. In fact, using the mixed method the number of unknowns corresponds to the sum of the number of elements and edges in which the physical domain has been discretized. Conversely in traditional methods the number of unknowns corresponds to either the number of element or nodes (FDM / FVM and FEM, respectively). This important drawback was one of the arguments used against mixed methods in the early works of Cordes & Kinzelbach (1992), Srivastava & Brusseau (1995).

Additionally, the discrete linear system obtained using the mixed formulation is indefinite and therefore, generally, not easy to solve. This issue was resolved by augmenting the discrete linear system by means of Lagrange multipliers, resulting in what is known as the mixed-hybrid finite element method (MHFEM) (Brezzi & Fortin 1991). The discrete linear system obtained by MHFEM is symmetric positive definite (SPD) and therefore can be easily solved using the conjugate gradient (CG) method. Furthermore, the size of the system of equations is reduced (to the number of edges) as the pressure and velocity unknowns are algebraically eliminated. Hence, just based on the size of the discrete linear system, the MHFEM is computationally less expensive than MFEM but still more costly than traditional methods.

Obviously, nowadays the computational cost is less of a problem than it was twenty years ago. In fact problems of the order of $10^6$ degrees of freedom can be easily solved on standard dual-core laptop PC with 4GB of RAM (see Chapter 3). Larger problems of the order of $10^7$ - $10^8$ unknowns require, in general, parallel computations independent of the method used for the approximation. Examples of parallel computation of groundwater flow in heterogeneous media can be found in Cliffe et al. (2000), de Dreuzy et al. (2007). If any existed, the concerns about CPU cost and time efficiency for the mixed methods have been overcome. Furthermore, considering that mixed methods provide a very accurate velocity solution and that this is of critical importance in many practical applications, the additional computational expense required to solve the linear systems obtained by mixed methods seems to be justified.

Although some of the limitations of mixed methods have been resolved, it is a matter of fact that these methods have not been frequently used in real-life applications and are not part of popular computer software such as MODFLOW (Harbaugh & McDonald 1996, Harbaugh et al. 2000) and FEFLOW (Diersch 1996), extensively used in the groundwater modelling community. In fact, the issue of accurate groundwater fluxes and locally conservative numerical methods is arguably unknown to practitioners who tend to develop groundwater models based on the approximation techniques deployed by commercially available softwares (generally FDM, FEM and FVM).

To the author's knowledge, it appears that a publicly available computer program (for groundwater modelling applications) based on mixed methods has not yet been developed. The programming codes currently existing such as PIFISS (Silvester & Powell 2007) or the MATLAB (MATLAB 1997) scripts of Bahriawati & Carstensen (2005) are a useful starting point and of great research interest. However they are far from being tools usable in applications. On the other hand, the mathematical theory underpinning the mixed formulation is well-developed and mature (see Raviart & Thomas (1977), Nedelec (1980), Arnold & Brezzi (1985), Chavent & Jaffré (1986), Roberts & Thomas (1991), Chavent & Roberts (1991), Brezzi & Fortin (1991), for example). Therefore, there exists a gap between the theory and the application.

In addition to mixed methods there are several other numerical techniques that are locally conservative and provide accurate approximations for the (groundwater) fluxes. A review of some of these techniques is given by Klausen & Russell (2004). The authors look at the relationship between traditional MFEM, control-volume mixed finite element method (CVMFEM) (Cai et al. 1997), enhanced cell-centered finite difference method (ECCFDM) (Arbogast et al. 1997, 1998) and multi-point flux approximation (MPFA) (Edwards & Rogers 1998, Aavatsmark, T., Bøe & Mannseth 1998*a*,*b*, Aavatsmark, T. & Mannseth 1998, Edwards 2002, Aavatsmark 2002, Edwards & Pal 2008, Edwards & Zheng 2008, Friis et al. 2008, Edwards & Zheng 2010, 2011). The study of locally conservative numerical methods is a very active area of research (see Edwards (2002) and all articles therein) and it is outside the scope of this chapter to review all the work which has been carried out on the subject.

Error and convergence analysis for the lowest order Raviart-Thomas ($RT_0$) mixed finite element method is well established (see Brezzi & Fortin (1991), Arbogast et al. (1996), Demlow (2002), Radu et al. (2004), for example). Similar papers are available for the MPFA method (Klausen & Winther 2006*b*,*a*, Klausen et al. 2008). Bause et al. (2010) compared the quality of the flux approximations of the two methods and showed that although the MFEM is slightly superior to MPFA, generally the two methods are qualitatively very similar. Crucially however, for MFEM approximations ($RT_0$ or Brezzi-Douglas-Marini $BDM_1$ (Brezzi et al. 1985)) existence and uniqueness of the discrete solution is guaranteed for any mesh (triangular type was considered in the paper) and diffusion full-tensor. The same has not yet been proven for MPFA methods on unstructured grids (Remark 3.1, Bause et al. (2010)). Existence and uniqueness on cell centred triangles is reported in Friis et al. (2008).

Similar studies have focused not only on error estimates for MFEM and MPFA but also on their computational cost. It should be said that the majority of the research focuses on the computational comparison of the hybrid version of the MFEM (the SPD version) with other techniques (see for example Kaasschieter & Huijben (1992), Younes et al. (1999), Younes & Fontaine (2008*b*,*a*)). Several studies have tried to link mixed formulations with standard finite volume methods with the aim of reducing the number of unknowns of MFEM (see Edwards (2002), Chavent et al. (2003), Younes et al. (2004), Brezzi et al. (2004), Edwards & Pal (2008), for example). Similarly, the link between MPFA and mixed methods is given in Vohralik (2006), Klausen & Russell (2004), Wheeler & Yotov (2006), Younes & Fontaine (2008*b*,*a*). The effort made in the last ten years or so to produce a numerical method which would give piecewise constant approximations for the pressure in each finite element and pressure dependent expressions for the fluxes has produced a large number of different numerical schemes.

In contrast, studies on the classical MFEM for which the associated discrete linear system is indefinite are significantly, less common (than the SPD version) and represent a somewhat specialist area of research. The saddle-point problem obtained from the mixed formulation can be solved using the minimal residual method (MINRES) (Paige & Saunders 1975). If MINRES is preconditioned with efficient symmetric preconditioners then the solution of the symmetric indefinite system can be very efficient (see Rusten & Wither (1992), Vassilevski & Lazarov (1996), Powell (2003), Powell & Silvester (2003), Powell (2005)). Of course, for MINRES to be competitive the choice of preconditioner is crucial.

It is clear that the essential prerequisites for the numerical methods used in this work are:

- Locally conservative (at the finite element level);
- Accurate in the computation of fluxes;
- Robust with respect to heterogeneous and discontinuous conductivity coefficient;
- Ideally also robust with respect to anisotropic conductivity coefficients as recently achieved with MPFA methods (Edwards & Zheng 2008, 2010, 2011).

It is equally clear that there are several methods that satisfy these conditions some of which have been mentioned in the previous section. However, it should be kept in mind that the main objective of this thesis is to study stochastic Galerkin (SG) methods and, as we will show in Chapter 4, these build from the approximation method used for deterministic problems. Therefore the choice of numerical methods to use in our (deterministic) work is intrinsically linked with the requirements of SG methods.

In methods such as MPFA, CVMFEM and ECCFDM the fluxes are obtained through explicit expressions which are functions of the pressure. The MFEM does not have such explicit expressions. Although the explicit representation of the fluxes can be advantageous, for example in the field of multiphase flow (Klausen & Russell 2004), it is not entirely certain that such a representation is possible in a stochastic framework. To the author's knowledge most of the work in the field of SG methods has used FDM, FEM and MFEM approximations for the deterministic operator. An extensive review of SG methods is given in Chapter 4, §4.1.

Given the aforementioned motivations, in this thesis we are concerned with classical MFEM formulations. In this chapter we review the theory of the MFEM including the hybrid formulation. This is standard material and extensive references have been provided throughout this introduction. A review of solution strategies for these methods is given in §2.5.3 and §2.6.1, respectively. Chapter 3 compares the computational efficiency of MFEM and MHFEM for a range of numerical examples. For the MFEM (indefinite case) we use practical preconditioners proposed in Powell (2003), Powell & Silvester (2003). For the MFEM (SPD case) we use an approximation of the coefficient matrix as a preconditioner for CG. The analysis includes test problems with full-tensor, anisotropic coefficients on structured and unstructured triangular / quadrilateral meshes.

## 2.2 The mathematical model

The steady-state flow of water in porous media is described by a scalar second-order partial differential equation, the solution of which, when supplemented with suitable boundary conditions, gives the distribution of a scalar potential $u$ (potential head) over a specific domain, $D$. Let $D$ be a domain in $\mathbb{R}^d$, $d = 2, 3$, bounded by $\Gamma = \Gamma_D \cup \Gamma_N$, where $\Gamma_D$ and $\Gamma_N$ denote the portion of $\Gamma$ where Dirichlet and Neumann boundary conditions are prescribed, respectively. We seek a solution $(u)$

to the second-order elliptic problem

$$-\nabla \cdot \mathcal{C}\nabla u = f(\mathbf{x}) \qquad \text{in } D, \tag{2.1a}$$

$$u = g(\mathbf{x}) \qquad \text{on } \Gamma_D, \tag{2.1b}$$

$$\mathcal{C}\nabla u \cdot \mathbf{n} = 0 \qquad \text{on } \Gamma_N, \tag{2.1c}$$

where $\mathcal{C}$ is a $d \times d$ symmetric positive definite coefficient tensor representing the hydraulic conductivity specific to the problem at hand, $\mathbf{n}$ denotes the unit outward normal vector to $\Gamma_N$ and $g(\mathbf{x})$ represents the prescribed constant head on $\Gamma_D$. $f(\mathbf{x})$ represents a sink or source term.

Traditionally, finite difference or finite element methods are used to discretise problem (2.1). In such methods it is common to post-process the approximation to the potential, $u$, to obtain the fluid discharge (flux) or velocity, $\mathbf{q}$, according to Darcy's Law. Whilst this is commonly done, many authors have shown that the computed fluxes are not accurate due to errors introduced by numerical differentiation (see Mosé et al. (1994) and Kaasschieter & Huijben (1992), for example).

Very often, in applications, $\mathbf{q}$ is the variable of primary interest. Hence, a numerical scheme which guarantees an accurate approximation of the fluxes is required. This can be achieved re-stating problem (2.1) by explicitly introducing Darcy's Law. We now seek the simultaneous solution $(\mathbf{q}, u)$ to the coupled first-order problem

$$\mathcal{C}^{-1}\mathbf{q} - \nabla u = 0 \qquad \text{in } D, \tag{2.2a}$$

$$\nabla \cdot \mathbf{q} = -f(\mathbf{x}) \qquad \text{in } D, \tag{2.2b}$$

$$u = g(\mathbf{x}) \qquad \text{on } \Gamma_D, \tag{2.2c}$$

$$\mathbf{q} \cdot \mathbf{n} = 0 \qquad \text{on } \Gamma_N. \tag{2.2d}$$

The solution of problem (2.2) using mixed finite element methods allow us to obtain a simultaneous approximation for the potential and the flux everywhere in $D$.

In the next section an outline of the mixed finite element theory is given, which, although somewhat technical, is needed for a complete presentation of the subject.

## 2.3 Preliminary Definitions

The notions reported in this section are standard and well accepted definitions and follow the rigorous analysis originally reported in Brezzi & Fortin (1991) and subsequent works such as in Powell (2003). These definitions, although available in those references, are included in this thesis as they form the basis of our analysis and derivation of the weak formulation for the first order problem (2.2).

Let $D$ be a bounded and connected domain in $\mathbb{R}^{\mathcal{D}}$, $\mathcal{D} = 2, 3$, with Lipschitz continuous boundary $\Gamma = \Gamma_D \cup \Gamma_N$.

Define the Lebesgue space $L^2(D)$ of scalar square integrable functions over $D$,

$$L^2(D) = \{w : w \text{ is defined on } D \text{ and} \int_D w^2 dD < \infty\}. \tag{2.3}$$

$L^2(D)$ is a Hilbert space with the inner product

$$(w, s) = \int_D ws\ dD,$$

and associated norm

$$\|w\|_{L^2(D)} = \left(\int_D w^2 dD\right)^{\frac{1}{2}} = (w, w)^{\frac{1}{2}}. \tag{2.4}$$

Similarly, for vector functions $\mathbf{v} = (v_1, \ldots, v_d)^T$, we define the Lebesgue space, $L^2(D)^d$, of vectorial square integrable functions over $D$,

$$L^2(D)^d = \{\mathbf{v} : v_i \ \in \ L^2(D),\ i = 1, \ldots, d\}. \tag{2.5}$$

$L^2(D)^d$ is a Hilbert space with the inner product,

$$(\mathbf{v}, \mathbf{u}) = \int_D \mathbf{v} \cdot \mathbf{u}\ dD = \sum_{i=1}^{d} \int_D v_i u_i dD,$$

and associated norm,

$$\|\mathbf{v}\|_{L^2(D)^d} = \left( \int_D \mathbf{v}^2 dD \right)^{\frac{1}{2}} = (\mathbf{v}, \mathbf{v})^{\frac{1}{2}} . \tag{2.6}$$

In order to derive the weak variational formulation of problem (2.2), we need to define the following Sobolev space

$$H^1(D) = \{w : w \in L^2(D) \text{ and } \frac{\partial w}{\partial x_i} \in L^2(D), i = 1, \ldots, d\}. \tag{2.7}$$

This is a Hilbert space with inner product,

$$(w, s) = \int_D (ws + \nabla w \cdot \nabla s) \;\; dD,$$

and associated norm,

$$\|w\|_{H^1(D)} = \left( \int_D w^2 + |\nabla w|^2 \; dD \right)^{\frac{1}{2}} . \tag{2.8}$$

A well known subspace of $H^1(D)$ is the subspace $H^1_0(D)$ of functions that vanish at the boundary $\Gamma$ of $D$,

$$H^1_0(D) = \{w \in H^1(D) : w = 0 \text{ on } \Gamma\}. \tag{2.9}$$

Functions belonging to $H^1_0(D)$ satisfy the *Poincaré-Friedrich's inequality* (see Braess (1992) for definition and proof), thus ensuring uniqueness of the solution. The set of functions vanishing on the Dirichlet portion of $\Gamma$ belong to the Hilbert space

$$H^1_{0,D}(D) = \{w \in H^1(D) : w = 0 \text{ on } \Gamma_D\}. \tag{2.10}$$

The following Hilbert spaces are required for the mixed variational formulation of problem (2.2). Define the space $H(div; D)$ of square-integrable vectorial functions whose divergences are also square-integrable

$$H(div; D) = \{\mathbf{v} : \mathbf{v} \in L^2(D)^d \text{ and } \nabla \cdot \mathbf{v} \in L^2(D)\}, \tag{2.11}$$

this space is equipped with the inner product,

$$(\mathbf{v},\mathbf{u})_{div} = (\mathbf{v},\mathbf{u}) + (\nabla\cdot\mathbf{v},\nabla\cdot\mathbf{u}),$$

and associated norm,

$$\|\mathbf{v}\|_{div} = \|\mathbf{v}\|_{L^2(D)^d} + \|\nabla\cdot\mathbf{v}\|_{L^2(D)}. \tag{2.12}$$

Define $w_\Gamma$, to be the trace of any scalar function $w \in H^1(D)$. Thus the set of all traces determines the following Hilbert space

$$H^{\frac{1}{2}}(\Gamma) = \{g : g = w_\Gamma \text{ for some } w \in H^1(D)\}. \tag{2.13}$$

Similarly, for vectorial functions $\mathbf{v} \in H(div; D)$, $(\mathbf{v}\cdot\mathbf{n})_\Gamma$ defines the normal trace, where $\mathbf{n}$ is the normal outward pointing unit vector to $\Gamma$. Therefore the set of all such functions determines

$$H^{-\frac{1}{2}}(\Gamma) = \{q : q = (\mathbf{v}\cdot\mathbf{n})_\Gamma \text{ for some } \mathbf{v} \in H(div; D)\}. \tag{2.14}$$

Following Powell (2003), for any function $g \in H^{\frac{1}{2}}(D)$ and $q \in H^{-\frac{1}{2}}(D)$, $\langle\cdot,\cdot\rangle$ represents the duality pairing

$$\langle g, q\rangle = \int_\Gamma gq\ ds, \tag{2.15}$$

and we can define and important subspace of $H(div; D)$ in which the solution for the flux and / or velocity $\mathbf{q}$ is sought

$$H_{0,N}(div; D) = \{\mathbf{v} \in H(div; D) : \langle\mathbf{v}\cdot\mathbf{n}, w\rangle = 0 \qquad \forall w \in H^1_{0,D}(D)\}. \tag{2.16}$$

## 2.4 Continuous Weak Form

Define $W = L^2(D)$ and $V = H(div; D)$. Multiplying (2.2b) by a scalar basis function $w \in W$ and integrating over $D$ yields

$$\int_D (\nabla\cdot\mathbf{q})w\ dD\ = -\int_D fw\ dD.$$

Define the bilinear form $b(\cdot,\cdot)$ and the linear functional $L(\cdot)$ by

$$\begin{aligned} b(\mathbf{q},w) &= \int_D (\nabla\cdot\mathbf{q})w\ dD, \\ L(w) &= \int_D fw\ dD \equiv (f,w)_{L^2(D)} \end{aligned} \tag{2.17}$$

Now multiply (2.2a) by a vectorial basis function $\mathbf{v}\in V$ and integrate over $D$ to give

$$\int_D \mathcal{C}^{-1}\mathbf{q}\cdot\mathbf{v}dD - \int_D \nabla u\cdot\mathbf{v}\ dD = 0.$$

Using the following version of Green's formula (see Brezzi & Fortin (1991))

$$\int_D \nabla\cdot\mathbf{v}wdD = -\int_D \mathbf{v}\cdot\nabla wdD + \int_\Gamma (\mathbf{v}\cdot\mathbf{n})\,wd\Gamma \qquad \forall w\in H^1(D) \tag{2.18}$$

we obtain the following bilinear forms

$$\begin{aligned} a(\mathbf{q},\mathbf{v}) &= \int_D \mathcal{C}^{-1}(\mathbf{q}\cdot\mathbf{v})\ dD, \\ b(\mathbf{v},w) &= \int_D \nabla\cdot\mathbf{v}w\ dD. \end{aligned}$$

Finally, the weak formulation of the mixed variational problem (2.2) is : find $(\mathbf{q},u)\in V\times W$ such that

$$\begin{aligned} a(\mathbf{q},\mathbf{v})+b(\mathbf{v},u) &= \langle g,\mathbf{n}\cdot\mathbf{v}\rangle_{\Gamma_D} && \forall\mathbf{v}\in V \\ b(\mathbf{q},w) &= -(f,w) && \forall w\in W. \end{aligned} \tag{2.19}$$

The weak formulation has a unique solution $(\mathbf{q},u)\in V\times W$ provided the bilinear forms $a\left(\cdot,\cdot\right)$ and $b\left(\cdot,\cdot\right)$ satisfy the following inf-sup condition (also called the Ladyzhenskaya-Babuška-Brezzi (LBB) condition)

$$\inf_{w\in W}\sup_{\mathbf{v}\in V}\frac{\displaystyle\int_D w\nabla\cdot\mathbf{v}}{\parallel w\parallel_W \parallel \mathbf{v}\parallel_V} \geq \beta, \tag{2.20}$$

where the constant $\beta>0$ (for a proof of this condition see Brezzi & Fortin (1991)).

## 2.5 Mixed Finite Element Approximation

Let $T^h$ be a partition of $D$ defined by closed sub-domains, *finite elements*, $K_i, i = 1, \ldots, n$, such that,

$$T^h = \bigcup_{k=1}^{n} K_k$$

where $h$ denotes the discretisation parameter which describes the size of the finite elements in $T^h$. Let $E^h$ be the collection of numbered edges ($\mathcal{D} = 2$) or faces ($\mathcal{D} = 3$), $e_i$, $i = 1, \ldots, m$, where $m$ is the total number of edges in $T^h$. According to the Galerkin method we define the finite dimensional subspaces $V^h \subset V$ and $W^h \subset W$. The discrete variational formulation of (2.19) is: Find $(\mathbf{q}^h, u^h) \in V^h \times W^h$ such that

$$\begin{aligned} a\left(\mathbf{q}^h, \mathbf{v}^h\right) + b\left(\mathbf{v}^h, u^h\right) &= \langle g, \mathbf{n} \cdot \mathbf{v}^h \rangle_{\Gamma_D} \qquad \forall \mathbf{v}^h \in V^h \\ b\left(\mathbf{q}^h, w^h\right) &= -\left(f, w^h\right) \qquad \forall w^h \in W^h \end{aligned} \tag{2.21}$$

### 2.5.1 Raviart-Thomas Approximation

A family of local spaces that can be used to construct a suitable subspace $V^h \subset V \equiv H_{0,N}(div; \Omega)$ was proposed by Raviart & Thomas (1977) for $\mathbb{R}^2$ and by Nedelec (1980) for $\mathbb{R}^3$. Let $RT^0$ denote the space of linear vectorial functions $\mathbf{v}_i$, $i = 1, \ldots, I$, where $I$ is the number of edges or faces associated with a finite element $K$. Therefore, we have

$$RT^0(K) = span\{\mathbf{v}_i\}_{i=1}^{I}.$$

The value of $I$ depends on the type of finite element chosen for the discretisation of $D$, so that $I = 3$ and $I = 4$ for triangular and rectangular elements, respectively, and $I = 4$ and $I = 6$ for tetrahedra and parallelepipeda, respectively.

It is common practice to define the vectorial basis functions on a reference element $\hat{K}$. Thus the definition of vectorial basis functions on a general element follows from the reference element through an affine transformation. In such circumstances the well-known transformation rules for vectorial and scalar basis functions apply (see Brezzi & Fortin (1991), §III.1.3). Let $RT^0(\hat{K})$ denote the local $I$-dimensional space of vectorial basis functions $\hat{\mathbf{v}}_i$ defined on $\hat{K}$. It follows that

$$RT^0(K) = \left\{ \mathbf{v} : \mathbf{v}(\mathbf{x}) = \frac{\mathbf{B}\hat{\mathbf{v}}(\boldsymbol{\xi})}{J} \;\; \forall\, \boldsymbol{\xi} \in \hat{K} \text{ and } \hat{\mathbf{v}} \in RT^0(\hat{K}) \right\}, \tag{2.22}$$

where $\boldsymbol{\xi}$ is the local coordinate system and $J$ is the determinant of the Jacobian of the transformation $\mathbf{B}$. We can now define the global spaces

$$RT^0(D;T^h) = \{\mathbf{v} \in H(div;D) : \mathbf{v}|_K \in RT^0(K) \;\forall K \in T^h\}, \tag{2.23}$$

and

$$\mathcal{M}^0 = \left\{ \mathbf{v} \in L^2(D)^d \text{ and } \mathbf{q}|_K \in RT^0(K) \;\forall K \in T^h \right\}. \tag{2.24}$$

A suitable subspace for the approximation to the flux $\mathbf{q}$ is

$$V^h = \mathcal{M}^0 \cap H_{0,N}(div;D) = \left\{ \mathbf{v} \in RT^0(D;T^h) \text{ and } \mathbf{v}\cdot\mathbf{n}|_{\Gamma_N} = 0 \right\}. \tag{2.25}$$

For triangular and tetrahedra elements the vectorial basis functions $\hat{\mathbf{v}} \in RT^0(\hat{K})$ have the special form

$$\hat{\mathbf{v}} = \begin{pmatrix} a + c\xi \\ b + c\eta \end{pmatrix}, \qquad \hat{\mathbf{v}} = \begin{pmatrix} a + c\xi \\ b + c\eta \\ e + c\zeta \end{pmatrix},$$

respectively, and for rectangular and parallelepipeda elements the form

$$\hat{\mathbf{v}} = \begin{pmatrix} a + c\xi \\ b + d\eta \end{pmatrix}, \qquad \hat{\mathbf{v}} = \begin{pmatrix} a + c\xi \\ b + d\eta \\ e + f\zeta \end{pmatrix}.$$

respectively. The coefficients $a, b, c, d, e, \text{and} f$ are some constants chosen so that the integral of the normal component of $\hat{\mathbf{v}}$ on the edge or face of $\hat{K}$ is equal to some constant $\delta$.

Finally, the pressure $u$ is approximated by piecewise constant functions $w$. Let $M^0(K)$ denote the one-dimensional space of constant scalar basis functions on $K$. Hence, a suitable subspace $W^h \subset W \equiv L^2(D)$ is

$$W^h = \{w \in L^2(D) : w|_K \in M^0(K) \; \forall \; K \in T^h\}. \tag{2.26}$$

### 2.5.2 Linear System

For each element $K$ we associate a scalar basis function $\phi_j$, $i = 1, \ldots, n$, which is element-wise constant. The potential $u^h$ can therefore be approximated in terms of the global scalar basis functions,

$$u^h(\mathbf{x}) = \sum_{j=1}^{n} u_j \phi_j, \tag{2.27}$$

where $\phi_j$ is the characteristic function on $K_j$ i.e. it satisfies the following condition

$$\phi_j = \begin{cases} 1 & \text{if } \phi_j \in K_j \\ 0 & \text{elsewhere} \end{cases} \tag{2.28}$$

Globally, for each edge or face $e \in E^h$ we fix oriented normal vectors $\boldsymbol{\nu}_i$, $i = 1, \ldots, m$, where $m$ is the total number of edges in $E^h$. Now, we define a *direction* index $s_K^i$ so that

$$s_K^i = \begin{cases} +1 & \text{if } \mathbf{n}_K^i = \boldsymbol{\nu}_K^i \\ -1 & \text{if } \mathbf{n}_K^i = -\boldsymbol{\nu}_K^i \end{cases} \tag{2.29}$$

where $\mathbf{n}_K^i$ denotes the set of unit outward normal vectors at the edges $e_i \in E^h$.

The vectorial (flux) basis functions $\hat{\boldsymbol{\varphi}}_i \in V^h$ are defined with respect to a reference element $\hat{K}$ so that,

$$\int_{e_k} \hat{\boldsymbol{\varphi}}_i \cdot \hat{\boldsymbol{\nu}}_k ds = \begin{cases} 1 & \text{if } k = i \\ 0 & \text{if } k \neq i \end{cases} . \tag{2.30}$$

Note that this is the condition which ensures continuity of the normal components of the flux $\mathbf{q}$ across the interelement edges of $E^h$. Finally, we can approximate $\mathbf{q}^h$ in terms of the global vectorial basis functions $\boldsymbol{\varphi}_i$,

$$\mathbf{q}^h(\mathbf{x}) = \sum_{i=1}^{m} q_i \boldsymbol{\varphi}_i. \tag{2.31}$$

The mapping $\hat{\boldsymbol{\varphi}}_i \mapsto \boldsymbol{\varphi}_i$ follows from (2.22). Additionally, the global basis functions $\boldsymbol{\varphi}_i$ are multiplied by the index $s_i$ before the system is assembled. The source / sink term $f(\mathbf{x})$ is also approximated in terms of the global scalar basis functions $\phi_i$,

$$f(\mathbf{x}) \approx \sum_{i=1}^{n} f_i \phi_i. \tag{2.32}$$

Substituting expansions (2.27), (2.31) and (2.32) into (2.21) we obtain

$$\begin{aligned} \sum_{j=1}^{m} q_j A_{i,j} + \sum_{k=1}^{n} u_k B_{i,k} &= \mathbf{g} \\ \sum_{i=1}^{m} q_i B_{k,i} &= \mathbf{f} \end{aligned} \tag{2.33}$$

where $A_{i,j}$ is constructed from the element contributions

$$A_{i,j}^K = a\left(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i\right)_K = \int_K \mathcal{C}^{-1} \boldsymbol{\varphi}_j^K \boldsymbol{\varphi}_i^K dK, \qquad i,j = 1, \ldots, I, \tag{2.34}$$

where $I$ is the number of edges or faces on $K$. Given an element $K$, $\phi_K = 1$, hence the element contributions to the global matrix $B_{k,i}$ are given by

$$B_i^K = \int_{e_i} \boldsymbol{\varphi}_i \cdot \mathbf{n}_i \, de, \qquad i = 1, \ldots, I, \tag{2.35}$$

and

$$B_{k,i} = \begin{cases} 0 & \text{if } e_i \notin K_k \\ s_i^{K_k} & \text{if } e_i \in K_k \end{cases}. \tag{2.36}$$

The elements of the right-hand side vectors defined by

$$f_k = \int_{K_k} f dK_k \qquad g_i = \begin{cases} 0 & \text{if } e_i \notin \Gamma_D \\ \int_{e_i} g de & \text{if } e_i \in \Gamma_D \end{cases}. \tag{2.37}$$

The system (2.33) can be re-written in matrix notation as follows

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{f} \end{pmatrix}, \tag{2.38}$$

where $\boldsymbol{u} = [q_1, \dots, q_m]^T$ and $\boldsymbol{u} = [u_1, \dots, u_n]^T$. The matrix $A$ is generally referred to as a weighted velocity matrix and the matrix $B$ is considered to be a discrete representation of the divergence operator.

Given that $A$ is symmetric and positive definite, we have,

$$\mathbf{q} = A^{-1}\left(\mathbf{g} - B^T\mathbf{p}\right), \tag{2.39}$$

which if substituted into the second equation of (2.38) gives

$$BA^{-1}B^T\mathbf{p} = BA^{-1}\mathbf{g} - \mathbf{f}. \tag{2.40}$$

Matrix $BA^{-1}B^T$ is also symmetric and positive definite (see Chavent & Jaffré (1986) and Kaasschieter & Huijben (1992) for an alternative proof). This aspect is very important in terms of the choice of scheme to implement to solve the linear system (2.38).

### 2.5.3 Solution Strategies

A review of solution strategies for the linear system (2.38) is given by Scheichl (2000) and Powell (2003). The solution of system (2.40) by the conjugate gradient method is advocated by Kim (2001) and Ewing & Wheeler (1983). However, the computation of $A^{-1}$ is expensive for general meshes and full-tensor $\mathcal{C}$ and the Schur complement $BA^{-1}B^T$ is not sparse. When rectangular meshes and diagonal $\mathcal{C}$ are used the element contributions $A_K$ are block-diagonal (see Powell (2003)), hence the computation of $A_K^{-1}$ is cheap and the system (2.40) can be solved efficiently by CG. Additionally, it can be shown that if the trapezoidal quadrature rule is used

(Kaasschieter & Huijben (1992)) on rectangular meshes, $A$ becomes diagonal. For these special cases the solution of (2.40) using CG is recommended.

There have been various attempts to solve the saddle-point problem (2.38) using iterative methods. The Uzawa method is a well-suited iterative scheme to solve saddle-point systems. However, this method requires the computation of the inverse of the coefficient matrix which becomes infeasible for practical applications (unstructured meshes and full-tensor coefficients). Fortin & Glowinski (1983) introduced the augmented Lagrangian method which applies an Uzawa algorithm to a modified saddle-point problem.

Algebraic approaches to solve (2.38) were introduced by Rusten & Wither (1992) and several preconditioners are proposed in Rusten & Wither (1993) and Rusten et al. (1996). Powell (2003) and Powell & Silvester (2003) proposed an ideal and practical preconditioner of the form

$$P = \begin{pmatrix} diag(A) & 0 \\ 0 & B\,diag(A)^{-1}B^T \end{pmatrix}. \tag{2.41}$$

The Schur complement $B\,diag(A)^{-1}B^T$ can be solved exactly or approximated by one V-cycle of black-box Algebraic Multi-Grid (AMG).

We recall that a preconditioner is defined to be $h$-optimal when the solver iteration count is independent or almost independent of the discretisation parameter $h$. Powell & Silvester (2003) showed that the preconditioner defined by (2.41) is $h$-optimal for isotropic $\mathcal{C}$ on structured triangular and rectangular meshes. However $h$-optimality is lost for diagonal anisotropic coefficients on triangular meshes. Furthermore, (2.41) is never $h$-optimal for general full-tensor coefficients.

The definition of $\mathcal{C}$-optimality follows from above. The preconditioner (2.41) is $\mathcal{C}$-optimal, but only for some special cases. In fact, its efficiency decreases drastically for anisotropic diagonal and full tensor coefficients on structured triangular meshes.

For structured rectangular meshes (2.41) is more efficient showing $\mathcal{C}$ optimality also for anisotropic diagonal coefficients. Currently, a preconditioner for (2.38) which is $\mathcal{C}$-optimal for anisotropic full tensor coefficients has not yet been found.

Furthermore, the efficiency of (2.41) has not yet been tested on unstructured two-dimensional meshes and structured and unstructured three-dimensional meshes.

An approach which has been extensively used in the literature (see Kaasschieter (1995)) is the hybrid method, introduced by Fraeijs de Veubeke (1965) and further developed by Arnold & Brezzi (1985) and Brezzi & Fortin (1991). This is discussed further in the next section.

## 2.6 Mixed Hybrid Finite Element Method

Arnold & Brezzi (1985) presented a way to derive a symmetric positive definite coefficient matrix for problem (2.2). The continuity condition on the normal components of the flux $\mathbf{q}$ across the finite element edges or faces is relaxed, i.e. $\mathbf{q}$ is now discontinuous across element interfaces. The continuity condition (required for the type of problems herein investigated) is subsequently re-established by introducing Lagrange multipliers $\lambda$ at those interfaces. The velocity space, being discontinuous, can be eliminated obtaining a system with unknowns $u^h$ and Lagrange multipliers $\lambda^h$. Note that the Lagrange multipliers are themselves the solution for the potential $u$ at the element interfaces. Furthermore the unknowns $u^h$ can also be eliminated to obtain a system of equations depending only on the Lagrange multipliers $\lambda^h$. This final system is positive-definite and of size $m \times m$, where $m$ is the number of edges or faces in $T^h$. Hence the conjugate gradient can be used to solve the discrete linear system efficiently.

In the following discussion we use the notation of Brezzi & Fortin (1991). Let

$\Lambda_0(e)$ denote the space of constant functions on $e$, $\forall e \in E^h$. We define the multiplier space

$$\Lambda_0\left(E^h\right) = \left\{\lambda^h : \lambda^h|_e \in \Lambda_0(e) \forall e \in E^h\right\}, \tag{2.42}$$

and the subspaces of multipliers that either vanish or approximate $g$ on $\Gamma_D$

$$\begin{aligned} \Lambda_{0,\Gamma_D} &= \left\{\lambda \in \Lambda\left(E_h\right) : \lambda = 0 \text{ on } \Gamma_D\right\}, \\ \Lambda_{g,\Gamma_D} &= \left\{\lambda \in \Lambda\left(E_h\right) : \lambda = g^h \text{ on } \Gamma_D\right\}, \end{aligned} \tag{2.43}$$

where

$$\int_e \left(g^h - g\right) ds = 0, \qquad \forall e \in \Gamma_D.$$

The flux approximation $\mathbf{q}^h$ is now sought in $\mathcal{M}^0$ and the Lagrange multipliers are defined in $\Lambda_0(e)$. Hence the following bilinear forms are defined

$$\begin{aligned} c\left(\mu^h, \mathbf{q}^h\right) &= \sum_{K \in T^h} \int_{\Gamma_K} \mu^h \mathbf{q}^h \cdot \mathbf{n} \, d\Gamma_K \\ b\left(\mathbf{q}^h, w^h\right)_h &= \sum_{K \in T^h} \int_K \left(\nabla \cdot \mathbf{q}^h\right) w^h \, dK \end{aligned} \tag{2.44}$$

The hybrid version of the lowest-order Raviart-Thomas mixed method for problem (2.2) reads: Find $(\mathbf{q}^h, u^h, \lambda^h) \in \mathcal{M}^0 \times W^h \times \Lambda_{0,\Gamma_D}$ such that

$$\begin{aligned} a\left(\mathbf{q}^h, \mathbf{v}^h\right) + b\left(\mathbf{v}^h, u^h\right)_h &= c\left(\lambda_h, \mathbf{v}^h\right), \quad \forall \mathbf{v}^h \in \mathcal{M}^0, \\ b\left(\mathbf{q}^h, w^h\right)_h &= -\left(f, w^h\right), \quad \forall w^h \in W^h, \\ c\left(\lambda^h, \mathbf{q}^h\right) &= 0, \qquad\qquad \forall \lambda^h \in \Lambda_{0,\Gamma_D}. \end{aligned} \tag{2.45}$$

Given the space $\mathcal{M}^0$ as defined in 2.24 and the vectorial basis functions defined in §2.5.1, the approximation for the flux, $\mathbf{q}^h(\mathbf{x})$, can be expressed as follows

$$\mathbf{q}^h(\mathbf{x}) = \sum_{K \in T^h} \sum_{i=1}^{I^K} q_i^K \boldsymbol{\varphi}_i^K, \tag{2.46}$$

where, $I = 3, 4$ depending on the choice of finite elements for the discretisation of $T^h$.

The potential $u(\mathbf{x})$ is approximated as in (2.27). Before expressing the approximation for the Lagrange multipliers, let $\mathcal{I}^h \subset E^h$ be the collection of numbered edges ($\mathcal{D} = 2$) or faces ($\mathcal{D} = 3$), $e_i$, $i = 1, \ldots, l$, of $\{e \in E^h : e \not\subset \Gamma_D\}$. $l$ denotes the total number of edges in $\mathcal{I}^h$. The space $\Lambda_{0,\Gamma_D}$ is spanned by scalar basis functions $\mu_i$, $i = 1, \ldots, l$ that satisfy the following condition

$$\mu_i = \begin{cases} 1 & \text{if } e_i \in \mathcal{I}^h \\ \\ 0 & \text{elsewhere} \end{cases} . \tag{2.47}$$

The approximation of the Lagrange multipliers, $\lambda^h$, can now be stated as follows

$$\lambda^h(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i \mu_i. \tag{2.48}$$

Problem (2.45) can be re-stated in matrix notation as follows

$$\begin{pmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{f} \\ \mathbf{0} \end{pmatrix} . \tag{2.49}$$

The clear distinction between (2.49) and (2.21) is the choice of the approximation space for the flux $\mathbf{q}$. The space $\mathcal{M}^0$ does not require the continuity condition $\mathbf{q}^h \cdot \mathbf{n}$ which characterizes the space $V^h$ and in a more general sense the spaces $H(div; D)$. The basis for $\mathcal{M}^0$ is chosen so that $\mathbf{v}^h|_K \neq 0$ only in $K$ and vanishes elsewhere. The important advantage of defining $\mathbf{v}^h$ in a discontinuous space is that the matrix $A$ becomes block-diagonal and $\mathbf{q}$ can be eliminated at the element level as follows

$$\mathbf{q} = A^{-1}\left(\mathbf{g} - B^T\mathbf{u} - C^T\boldsymbol{\lambda}\right) \tag{2.50}$$

Note that, inverting $A$ corresponds to invering its diagonal blocks, thus this can be carried out at the element level with little computational expense. Now, using (2.50) to eliminate $\mathbf{q}$ from (2.49) we obtain the following system

$$\begin{pmatrix} BA^{-1}B^T & BA^{-1}C^T \\ CA^{-1}B^T & CA^{-1}C^T \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} BA^{-1}\mathbf{g} - \mathbf{f} \\ CA^{-1}\mathbf{g} \end{pmatrix}. \tag{2.51}$$

Now the matrix $BA^{-1}B^T$ is symmetric and positive definite (see Brezzi & Fortin (1991) and Kaasschieter & Huijben (1992) for the proof) and also diagonal (Kaasschieter & Huijben 1992). Therefore, we can eliminate the unknown $\mathbf{u}$ and obtain

$$\mathbf{u} = \left(BA^{-1}B^T\right)^{-1}\left(BA^{-1}\mathbf{g} - BA^{-1}C^T\boldsymbol{\lambda} - \mathbf{f}\right). \tag{2.52}$$

Using (2.52) in (2.51) we obtain the discrete linear system

$$D\boldsymbol{\lambda} = \mathbf{r}, \tag{2.53}$$

where

$$D = CA^{-1}C^T - CA^{-1}B^T\left(BA^{-1}B^T\right)^{-1}BA^{-1}C^T \tag{2.54}$$

and

$$\mathbf{r} = CA^{-1}\mathbf{g} + CA^{-1}B^T\left(BA^{-1}B^T\right)^{-1}\left(\mathbf{f} - BA^{-1}\mathbf{g}\right). \tag{2.55}$$

The matrix $D$ is symmetric and positive definite, hence (2.53) can be solved using the conjugate gradient method.

### 2.6.1 Solution Strategies

As already anticipated the discrete linear system (2.53) can be solved using the CG solver. There is a vast number of choices for a preconditioner based on $D$ to be used with CG. Among those choices are simple Successive Over-Relaxation (SOR or / and the symmetric version SSOR) preconditioning, and incomplete factorisations of $D$ (ILU) (see Saad (2003)).

A performance analysis for (2.53) using an incomplete Cholesky factorization of $D$ is available in Kaasschieter & Huijben (1992). Several authors use CG for (2.53) equipped with various preconditioners. Younes & Fontaine (2008*b*) use the efficient Eisenstat's implementation (Eisenstat 1981) of CG. The numerical experiments reported do not show either $h$-optimality nor $\mathcal{C}$-optimality. To the best knowledge of the author, an efficient preconditioner for (2.53) is currently not available.

Multigrid methods for symmetric and positive definite systems have been largely studied, and theory, implementation and applications are available in standard reference books (see Briggs et al. (2000), Hackbush (2003), for example). Convergence results for multigrid methods for nonconforming finite elements are given in Brenner (1989, 1992) and Braess & Verfürth (1990). Further results and comparison with mixed methods are given in Chen (1996). Although numerical results presented in these works show $h$-optimality, analysis of the effect of $\mathcal{C}$ is not included. The effect of the conductivity coefficient on AMG convergence is considered in Powell (2003). However, results for unstructured and 3-dimensional meshes are not provided.

In this work, we follow the ideas presented in Powell (2003) approximating $D$ by one V-cycle of AMG as preconditioner for (2.53). We extend the analysis on unstructured meshes.

The efficient solution of problems (2.1) and (2.2) for full-tensor, highly anisotropic coefficients remains an open question. Some authors have used sparse direct solvers for this purpose. Recently Younes & Fontaine (2008*a*) demonstrated the efficiency of sparse direct solvers based on unifrontal/multifrontal methods (Davis & Duff 1997, 1999) to solve (2.53) on quadrilateral meshes. Comparison with iterative methods is not provided.

The efficiency of sparse direct solvers such as *UMFPACK* (Davis 2004) depends on the size of the problem. The general consensus is that sparse direct solvers are very efficient for 2-dimensional problems, but their performance deteriorates for 3-dimensional problems. Certainly the trade off at which sparse direct solvers become less efficient than iterative solvers is problem dependent. In relation to this work, experiments only using iterative solvers are reported.

# Chapter 3

# Mixed and Hybrid Finite Element Numerical Experiments

## 3.1 Introduction

In this section the computational cost required to solve the linear systems obtained by the MFE and MHFE discretisations derived in Chapter 2 is evaluated. We use state-of-the-art iterative solvers equipped with efficient preconditioners. The computational cost is evaluated based on number of iterations $N_{it}$, required by the solver to achieve convergence, and the CPU time $t_{CPU}$ in seconds.

The codes herein deployed have been developed within the MATLAB environment (MATLAB 1997) and the computations are all performed in serial. The development of the same algorithms in a parallel architecture is matter for future work and development. The implementation of the Preconditioned Conjugate Gradient (PCG) algorithm follows Saad (2003) and the MINRES implementation was modified from Fischer (1996). The tolerance within the solvers is set to $10^{-10}$ and the maximum number of iterations is set to $maxit = 10^4$. All numerical experiments have been carried out using a standard dual-core laptop PC with 4GB of RAM.

The scope of this chapter can be summarised with the following question: Is solving the hybrid problem more efficient than solving the indefinite system generated by the mixed method? Given the review on solution strategies (see §2.5.3 and §2.6.1), it appears that the answer to this question is strictly dependent on the problem being considered.

Therefore, several test problems, each differing in terms of the conductivity coefficient $\mathcal{C}$, will be analysed. Numerical simulations are carried out on structured/unstructured triangular and rectangular meshes to assess the effect of discretisations on the solvers' performance. Throughout the discussion, emphasis is given to those problem settings where $h$ and $\mathcal{C}$ optimality is achieved.

Two tables are presented for each test problem. The first table includes results for preconditioned MINRES using (2.41) with a direct solver for the Schur complement. The preconditioned CG solver is used for the MHFE formulation (2.49) using an incomplete Cholesky factorisation of the matrix $D$ as preconditioner. These solvers are referred to as $p-MINRES$ and $PCG$ in the tables and following sections, respectively. In the second table results are presented for MINRES with one V-cycle of black-box AMG used for the approximation of the Schur complement. The preconditioner for CG is the AMG approximation of the coefficient matrix $D$. These solvers are referred to as $p-MINRES_{AMG}$ and $PCG_{AMG}$ in the tables and following sections, respectively.

The AMG solver we use is publicly available from the PIFISS (Silvester & Powell 2007) solvers library, written in MATLAB. Other versions written in FORTRAN / MATLAB such as the $HSL\_MI20$ (Boyle et al. 2007, 2009) are also freely available for academic use. Two types of smoothing functions are available in the library, these are the point Gauss-Seidel (PGS) and the point damped Jacobi (PDJ). In the following experiments we use the latter with two sweeps per iteration. Note that there is no attempt at tuning the several AMG parameters and that experiments with PGS were not carried out.

Note that the setup time for some of the preconditioners used in this chapter can be significantly large especially for fine meshes. In the tables included in the following sections the setup time has been reported as well as the solvers' solution timings.

The author would like to express his gratitude to Professor E.F. Kaasschieter for his help with the computer implementation of the MHFEM and for providing useful MATLAB functions to develop the code used for the experiments presented in this chapter and in this work in general.

## 3.2 Numerical experiments on triangular meshes

The numerical experiments are carried out on square domains. Structured meshes are obtained by partition of $D$ into regular squares of area $h^2$. Each square is further subdivided into two right angled triangles. Unstructured meshes are created by perturbation of structured meshes as explained in §3.2.5.

The analytical and numerical solutions for each test problem are presented. However, given that the MFE and MHFE solutions are equivalent, we only show results for the former method. The same applies for the potential and velocity $L^2$-norm error estimates. The $L^2$ error estimates are given by

$$\|\mathbf{q}-\mathbf{q}^h\|_{L^2} = \left\{ \sum_{i=1}^{T} |T_i| \left(\mathbf{q}_i - \mathbf{q}_i^h\right)^2 \right\}^{\frac{1}{2}}, \tag{3.1}$$

$$\|\phi-\phi^h\|_{L^2} = \left\{ \sum_{i=1}^{T} |T_i| \left(\phi_i - \phi_i^h\right)^2 \right\}^{\frac{1}{2}}, \tag{3.2}$$

where $|T_i|$ is the area of the finite element and $\mathbf{q}$ is evaluated at the centroid of each finite element using Darcy's Law. The numerically computed fluxes (normal components of the flux at the edge mid-sides) are post-processed to obtain values for $\mathbf{q}^h = (q_x, q_y)^h$ at each element centroid. The analytical and numerical potential solution is evaluated at the centroid of each finite element.

The same experiments presented in this section are reported for structured and unstructured rectangular meshes in §3.3.

### 3.2.1 Problem 1: heterogeneous, isotropic and diagonal $\mathcal{C}$

The first test problem is similar to the one presented in Kaasschieter & Huijben (1992). The conductivity coefficient is isotropic but heterogeneous (i.e. it varies spatially) and it is given by

$$\mathbf{K} = \begin{bmatrix} a(\mathbf{x}) & 0 \\ 0 & a(\mathbf{x}) \end{bmatrix},$$

where

$$a(\mathbf{x}) = \frac{1}{1 + 2\epsilon\cos(\pi x)\cos(\pi y) + \epsilon^2\cos^2(\pi y)}. \tag{3.3}$$

The case of hydraulic conductivity with sudden jumps (discontinuous case) is reported in §3.2.4. Given a source term $f = 0$ and boundary conditions defined by

$$\begin{aligned} g_D(\mathbf{x}) &= \pi(1-y), \qquad \mathbf{x} \in \Gamma_D \\ \Gamma_D &= \{\mathbf{x} \in \Gamma : y = 0 \text{ or } y = 1\}, \end{aligned} \tag{3.4}$$

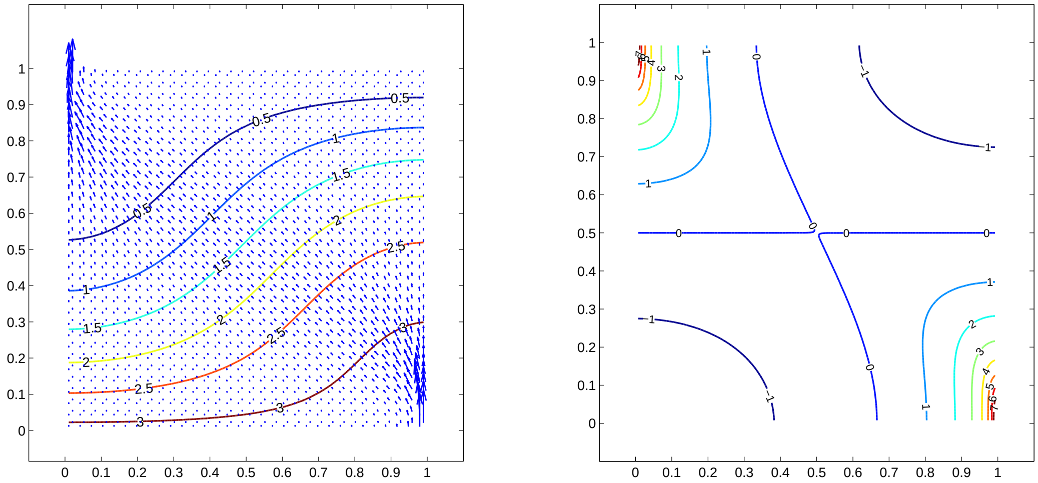
and

$$\begin{aligned} g_N(\mathbf{x}) &= 0, \qquad \mathbf{x} \in \Gamma_N \\ \Gamma_N &= \{\mathbf{x} \in \Gamma : x = 0 \text{ or } x = 1\}, \end{aligned} \tag{3.5}$$

the boundary value problem (2.1) has potential and velocity analytical solutions given by

$$
\begin{aligned}
u(\mathbf{x}) &= \pi(1-y) - \epsilon\cos(\pi x)\sin(\pi y). \\
\mathbf{q}(\mathbf{x}) &= -a(\mathbf{x})\begin{pmatrix} \pi\epsilon\sin(\pi x)\sin(\pi y) \\ -\pi-\epsilon\cos(\pi x)\cos(\pi y) \end{pmatrix}.
\end{aligned}
\tag{3.6}
$$

The MFEM potential and velocity solutions for $h=\frac{1}{32}$ and $\epsilon=0.9$ are depicted in Figure 3.1.



(a) Potential $u$ and velocity $\mathbf{q}$ solutions

(b) Log of conductivity field $\mathcal{C}(\mathbf{x})$

Figure 3.1: Numerical solutions and conductivity field for $\epsilon=0.9$ - Test problem 1

Table 3.1 shows $L^2(D^h)$ error estimates for the potential and the $x$ and $y$ components of the velocity field. The error estimates are in agreement with results presented by other authors (Kaasschieter & Huijben 1992) and with theoretical results (Brezzi & Fortin 1991). Second order convergence, $\mathcal{O}(h^2)$, is observed for the potential solution and first order convergence, $\mathcal{O}(h)$, for the velocity solutions.

The conductivity coefficient varies from $(1-\epsilon)^{-2}$ to $(1+\epsilon)^{-2}$. When $\epsilon\to 1$, $a(\mathbf{x})$

becomes singular and therefore the rate of convergence of the potential and velocity solutions deteriorates significantly (see Table 3.2 and 3.3). In fact, for $\epsilon = 0.999$ the $y$ component of the velocity solution does not converge. An analysis of the error distribution for the velocity components reveals that this is concentrated in the upper left and lower right corners of the domain. This location corresponds to the regions where the highest variation in the coefficient $a(\mathbf{x})$ occurs (see Figure 3.1b). This limitation could be resolved with local mesh refinement for the upper-left and lower-right regions of the domain.

Table 3.1: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 1, $\epsilon = 0.9$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $4.34E-03$ | – | $2.44E-01$ | – | $6.27E-01$ | – |
| $\frac{1}{32}$ | $9.25E-04$ | 2.23 | $1.15E-01$ | 1.09 | $2.97E-01$ | 1.08 |
| $\frac{1}{64}$ | $2.25E-04$ | 2.04 | $5.72E-02$ | 1.00 | $1.41E-01$ | 1.07 |
| $\frac{1}{128}$ | $5.61E-05$ | 2.00 | $2.85E-02$ | 1.01 | $7.02E-02$ | 1.01 |
| $\frac{1}{256}$ | $1.40E-05$ | 2.00 | $1.42E-02$ | 1.00 | $3.51E-02$ | 1.00 |

Table 3.2: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 1, $\epsilon = 0.99$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.09E-02$ | – | $7.43E-01$ | – | $1.64E+00$ | – |
| $\frac{1}{32}$ | $3.17E-03$ | 1.78 | $5.52E-01$ | 0.43 | $1.64E+00$ | 0.00 |
| $\frac{1}{64}$ | $8.19E-04$ | 1.95 | $3.59E-01$ | 0.62 | $1.51E+00$ | 0.11 |
| $\frac{1}{128}$ | $1.74E-04$ | 2.24 | $1.87E-01$ | 0.94 | $1.20E+00$ | 0.34 |
| $\frac{1}{256}$ | $3.27E-05$ | 2.41 | $8.51E-02$ | 1.13 | $6.18E-01$ | 0.96 |

Table 3.3: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 1, $\epsilon = 0.999$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.22E-02$ | – | $8.28E-01$ | – | $1.91E+00$ | – |
| $\frac{1}{32}$ | $3.97E-03$ | 1.62 | $6.92E-01$ | 0.26 | $2.19E+00$ | $< 0$ |
| $\frac{1}{64}$ | $1.29E-03$ | 1.63 | $5.74E-01$ | 0.27 | $2.48E+00$ | $< 0$ |
| $\frac{1}{128}$ | $4.09E-04$ | 1.65 | $4.65E-01$ | 0.30 | $2.69E+00$ | $< 0$ |
| $\frac{1}{256}$ | $1.25E-04$ | 1.72 | $3.57E-01$ | 0.38 | $2.75E+00$ | $< 0$ |

The numerical experiments using Krylov subspace methods for problem 1 are reported in Table 3.4. The table includes the number of iterations required to attain convergence, $N_{it}$, and the solution timings. For the $CG$, the set-up time for the preconditioner, i.e the time required to perform the incomplete Cholesky factorisation of the coeffcient matrix, is reported separately.

The post-processing time (MHFEM only) whereby the potential and velocity solutions are obtained from the Lagrange multipliers solution should also be considered. However this is negligible if compared with the set-up and solution times reported in Table 3.4. In fact, for a fine mesh, $h = \frac{1}{256}$, the post-processing time is only 0.15 seconds.

The data reported in Table 3.4 can be summarised as follow:

1. MINRES, equipped with the Schur complement preconditioner (2.41) is $h$-optimal and $\mathcal{C}$-optimal, when $\mathcal{C}$ is an isotropic diagonal tensor;

2. CG using an incomplete Cholesky factorization of the coefficient matrix $D$ as preconditioner, is $\mathcal{C}$-optimal but not $h$-optimal. $N_{it}$ grows linearly with $h$ leading to large CPU times for fine meshes;

3. On average the PCG CPU cost per iteration is lower than that required for preconditioned MINRES. Although this is a significant advantage of PCG, it is the overall number of iterations $N_{it}$ which determines the total CPU cost $t_{CPU}$;

4. The results presented indicate that heterogeneity has no effect on the performance of preconditioned MINRES. Conversely, although relatively small, an increase of $N_{it}$ and consequently $t_{CPU}$ is recorded using PCG for either small or large values of $\epsilon$.

The numerical experiments using algebraic multigrid as preconditioner are presented in Table 3.5. The main results can be summarised as follow:

Table 3.4: Iteration count and timings (set-up+solution time) for $p-MINRES$ and $PCG$ - Test problem 1

| | | $p-MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\epsilon$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\epsilon = 0.999$ | 43 | 0.97 | 135 | $1.11 + 1.35$ |
| | $\epsilon = 0.99$ | 43 | 0.97 | 139 | $1.10 + 1.42$ |
| | $\epsilon = 0.9$ | 44 | 1.03 | 138 | $1.11 + 1.42$ |
| $\frac{1}{128}$ | $\epsilon = 0.999$ | 43 | 5.54 | 256 | $17.97 + 11.63$ |
| | $\epsilon = 0.99$ | 43 | 5.46 | 255 | $17.99 + 11.87$ |
| | $\epsilon = 0.9$ | 43 | 5.58 | 270 | $18.09 + 12.10$ |
| $\frac{1}{256}$ | $\epsilon = 0.999$ | 43 | 28.38 | 525 | $285.16 + 113.93$ |
| | $\epsilon = 0.99$ | 43 | 28.34 | 495 | $281.56 + 108.26$ |
| | $\epsilon = 0.9$ | 43 | 28.36 | 535 | $284.56 + 117.26$ |

1. Inverting the Schur complement by AMG is more efficient than using sparse direct solvers. This determines lower CPU times than recorded in Table 3.4 even though the number of MINRES iterations is larger;

2. The computational efficiency of the AMG precoditioner is partly nullified by the large CPU time required to construct the coarse grids. This CPU cost grows linearly with the mesh size;

3. CG solution times and iteration counts are significantly reduced when one V-cycle of AMG code is used to approximately invert the MHFEM coefficient matrix;

4. As for the MINRES case the efficiency of AMG is partly nullified by the large computational cost of constructing the coarse grids for the approximation. Note that the coarsening process implemented on the MHFEM linear system is twice as expensive as the one implemented on the Schur complement system;

5. Both AMG implementations are $h$-optimal and $\mathcal{C}$-optimal;

6. Heterogeneity has no effect on the performance of both solvers.

Table 3.5: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 1

| | | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\epsilon$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\epsilon = 0.999$ | 49 | $1.33 + 0.98$ | 9 | $1.93 + 0.13$ |
| | $\epsilon = 0.99$ | 51 | $1.37 + 1.05$ | 9 | $1.94 + 0.13$ |
| | $\epsilon = 0.9$ | 51 | $1.29 + 1.04$ | 9 | $1.95 + 0.13$ |
| $\frac{1}{128}$ | $\epsilon = 0.999$ | 51 | $8.04 + 4.21$ | 10 | $13.98 + 0.50$ |
| | $\epsilon = 0.99$ | 52 | $8.28 + 4.38$ | 9 | $14.10 + 0.43$ |
| | $\epsilon = 0.9$ | 51 | $8.04 + 4.28$ | 9 | $13.88 + 0.42$ |
| $\frac{1}{256}$ | $\epsilon = 0.999$ | 56 | $110.60 + 22.42$ | 9 | $269.54 + 2.20$ |
| | $\epsilon = 0.99$ | 56 | $109.60 + 22.62$ | 10 | $281.54 + 2.34$ |
| | $\epsilon = 0.9$ | 54 | $108.31 + 22.03$ | 9 | $268.47 + 2.29$ |

### 3.2.2 Problem 2: heterogeneous, anisotropic and diagonal $\mathcal{C}$

The second test problem considers an heterogeneous, anisotropic and diagonal tensor. The conductivity coefficient $\mathcal{C}(\mathbf{x})$ is given by

$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} \alpha x^2 + y^2 & 0 \\ 0 & x^2 + y^2 \end{bmatrix}. \tag{3.7}$$

The anisotropy degree of the conductivity field varies depending on the values of the coefficient $\alpha$. When $\alpha = 1$, the conductivity field is isotropic.

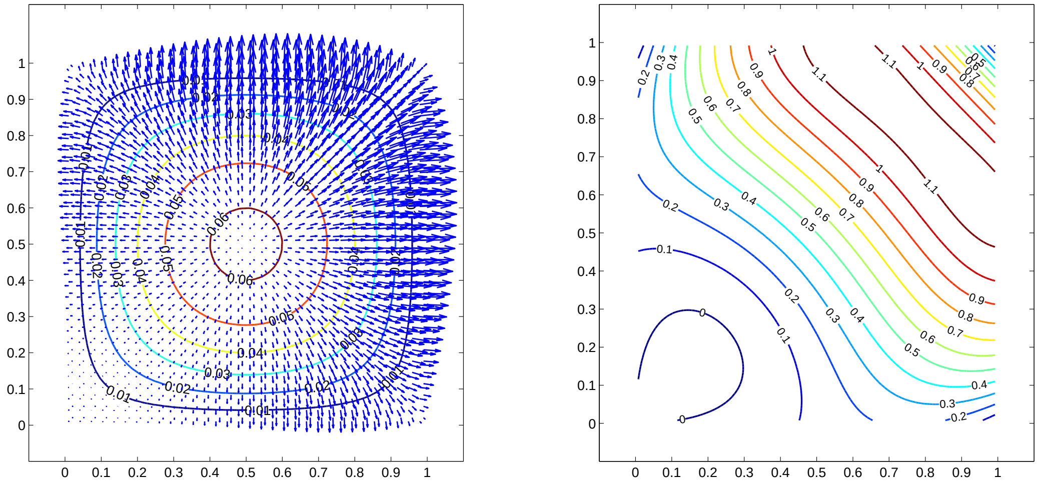
The potential and velocity analytical solutions are chosen so that homogeneous Dirichlet boundary conditions are prescribed on $\Gamma$. These are,

$$\begin{aligned} u(\mathbf{x}) &= (x - x^2)(y - y^2), \\ \mathbf{q}(\mathbf{x}) = &- \begin{pmatrix} (y^2 + x^2\alpha)(-1 + 2x)y(-1 + y) \\ (x^2 + y^2)x(-1 + x)(-1 + 2y) \end{pmatrix}. \end{aligned} \tag{3.8}$$

The source term is obtained by substituting (3.8) and (3.7) in (2.1), so that

$$\begin{aligned} f(\mathbf{x}) = -2x\alpha y + 2x\alpha y^2 + 6x^2\alpha y - 6x^2\alpha y^2 + \\ 2y^3 - 2y^4 - 2xy + 6xy^2 + 2x^2y - 6x^2y^2 + 2x^3 - 2x^4. \end{aligned} \tag{3.9}$$

The MFEM potential and velocity solutions for $\alpha = 1$ is depicted in Figure 3.2(a). The source term corresponding to (3.9) is illustrated in Figure 3.2(b).



(a) Potential $u$ and velocity $\mathbf{q}$ solutions

(b) Source term $f(\mathbf{x})$

Figure 3.2: MFEM solutions and source term - Test problem 2

Tables 3.6, 3.7 and 3.8 show $L^2(D^h)$ error estimates for the potential and the $x$ and $y$ components of the velocity field, for $\alpha = 10^{-2}, 1, 10^2$, respectively. As for the previous test case second order convergence, $\mathcal{O}(h^2)$, is recorded for the potential solution and first order convergence, $\mathcal{O}(h)$, for the velocity solutions. Note that, although the convergence rates are preserved for all values of $\alpha$, for the anisotropic case the absolute errors are two orders of magnitude larger for the potential solution and one order of magnitude larger for the velocity solutions when compared to the isotropic case.

The tables also include the minimum value for the potential solution, $u_{min}$. According to (3.8), $u(\mathbf{x})$ is always positive and it ranges from 0, at the domain boundaries, to 0.0625 at the center of the domain. Interestingly, for large anisotropic factors ($\alpha = 10^2$) unphysical negative oscillations in the potential solution are recorded (see

Table 3.8) for all values of $h$. The same behaviour is not recorded for small values of $\alpha$ (see Table 3.6).

Table 3.6: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 2, $\alpha = 10^{-2}$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate | $u_{min}$ |
|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.05E-03$ | – | $3.44E-03$ | – | $4.14E-03$ | – | $3.15E-04$ |
| $\frac{1}{32}$ | $2.80E-04$ | 1.91 | $1.73E-03$ | 0.99 | $2.09E-03$ | 0.99 | $8.02E-05$ |
| $\frac{1}{64}$ | $7.11E-05$ | 1.98 | $8.69E-04$ | 1.00 | $1.05E-03$ | 1.00 | $2.02E-05$ |
| $\frac{1}{128}$ | $1.78E-05$ | 2.00 | $4.35E-04$ | 1.00 | $5.24E-04$ | 1.00 | $5.07E-06$ |
| $\frac{1}{256}$ | $4.46E-06$ | 2.00 | $2.17E-04$ | 1.00 | $2.62E-04$ | 1.00 | $1.27E-06$ |

Table 3.7: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 2, $\alpha = 1$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate | $u_{min}$ |
|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.85E-04$ | – | $4.32E-03$ | – | $4.32E-03$ | – | $3.22E-04$ |
| $\frac{1}{32}$ | $4.65E-05$ | 1.99 | $2.18E-03$ | 0.99 | $2.18E-03$ | 0.99 | $8.11E-05$ |
| $\frac{1}{64}$ | $1.16E-05$ | 2.00 | $1.09E-03$ | 1.00 | $1.09E-03$ | 1.00 | $2.03E-05$ |
| $\frac{1}{128}$ | $2.90E-06$ | 2.00 | $5.47E-04$ | 1.00 | $5.47E-04$ | 1.00 | $5.08E-06$ |
| $\frac{1}{256}$ | $7.26E-07$ | 2.00 | $2.74E-04$ | 1.00 | $2.74E-04$ | 1.00 | $1.27E-06$ |

Table 3.8: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 2, $\alpha = 10^2$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate | $u_{min}$ |
|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $5.38E-03$ | – | $3.57E-01$ | – | $3.00E-01$ | – | $-8.10E-03$ |
| $\frac{1}{32}$ | $1.35E-03$ | 1.99 | $1.79E-01$ | 0.99 | $1.51E-01$ | 0.99 | $-2.19E-03$ |
| $\frac{1}{64}$ | $3.39E-04$ | 2.00 | $8.97E-02$ | 1.00 | $7.53E-02$ | 1.00 | $-5.65E-04$ |
| $\frac{1}{128}$ | $8.47E-05$ | 2.00 | $4.49E-02$ | 1.00 | $3.77E-02$ | 1.00 | $-1.43E-04$ |
| $\frac{1}{256}$ | $2.12E-05$ | 2.00 | $2.24E-02$ | 1.00 | $1.88E-02$ | 1.00 | $-3.58E-05$ |

The computational cost of solving the MFEM and MHFEM linear systems for diagonal anisotropic conductivity coefficients is reported in Tables 3.9 and 3.10.

Following the same structure used for test problem 1, Table 3.9 reports the computational cost of MINRES using the exact version of preconditioner (2.18). For the MHFEM system, CG is used in conjunction with an incomplete Cholesky factorisation of the coefficient matrix.

The numerical experiments were carried out with anisotropic coefficient $\alpha$ ranging from $10^{-2}$ to $10^2$. The main results reported in Table 3.9 are summarised as follows:

1. Anisotropy deteriorates the performance of both preconditioned MINRES and CG. The number of $p - MINRES$ iterations for $\alpha = 10^{-2}$ and $\alpha = 10^{2}$ is between five to six times larger than for the isotropic case;

2. For finer meshes ($h = \frac{1}{256}$) the factorisation of the coefficient matrix becomes increasingly costly, determining larger CPU costs than preconditioned MINRES;

3. In general, the solvers are not $\mathcal{C}$-optimal. However, fixing $\alpha$, MINRES is $h$-optimal. The CG iteration count varies largely also for fixed $\alpha$.

Table 3.9: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 2

| | | $p - MINRES$ | | PCG | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 240 | 5.93 | 78 | $0.99 + 0.92$ |
| | $\alpha = 1$ | 43 | 0.82 | 112 | $1.08 + 1.13$ |
| | $\alpha = 10^{-2}$ | 211 | 5.10 | 110 | $1.03 + 1.10$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 246 | 32.51 | 155 | $15.80 + 6.94$ |
| | $\alpha = 1$ | 43 | 5.40 | 219 | $17.45 + 10.03$ |
| | $\alpha = 10^{-2}$ | 226 | 29.66 | 225 | $16.39 + 10.38$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 248 | 166.29 | 313 | $242.93 + 67.91$ |
| | $\alpha = 1$ | 43 | 28.68 | 435 | $266.38 + 94.77$ |
| | $\alpha = 10^{-2}$ | 233 | 155.29 | 465 | $248.99 + 100.43$ |

The results for the numerical experiments using AMG as preconditioner for CG and MINRES are reported in Table 3.10. These can be summarised as follow:

1. In contrast to the isotropic case, the overall CPU cost (AMG coarsening and MINRES solution time) is lower than the exact version (see Table 3.9);

2. Similarly to test problem 1, the solution timings and iteration counts recorded for CG preconditioned by the AMG approximation of the coefficient matrix are by far the smallest among all methods considered. The AMG efficiency is partly nullified by the large cost of constructing the grids for the approximation.

This is twice as much as implementing the coarsening on the Schur complement system;

3. The experiments show that, for $\alpha \neq 1$, the number of CG iterations varies slightly with respect to the isotropic case. Conversely, the MINRES iteration count is between five to six times larger.

Table 3.10: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 2

| | | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 235 | 2.81 | 10 | $1.96 + 0.15$ |
| | $\alpha = 1$ | 50 | 0.61 | 9 | $1.96 + 0.13$ |
| | $\alpha = 10^{-2}$ | 212 | 2.51 | 11 | $2.04 + 0.15$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 242 | 10.56 | 12 | $12.75 + 0.56$ |
| | $\alpha = 1$ | 52 | 2.17 | 9 | $14.36 + 0.44$ |
| | $\alpha = 10^{-2}$ | 227 | 10.11 | 12 | $14.01 + 0.56$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 245 | 54.91 | 13 | $241.33 + 3.18$ |
| | $\alpha = 1$ | 54 | 11.73 | 10 | $256.18 + 2.43$ |
| | $\alpha = 10^{-2}$ | 232 | 52.73 | 12 | $251.11 + 2.88$ |

When the conductivity coefficient is a diagonal anisotropic tensor, MINRES preconditioned by (2.41) is not $\mathcal{C}$-optimal. The reason for this is associated with the structure of the element stiffness matrix. It can be shown, in fact, that each row of the element stiffness matrix (triangular elements) is scaled with respect to both coefficients of the diagonal tensor, $\mathcal{C}$ (see Powell (2003)). This causes a significant deterioration in MINRES performance and loss of $\mathcal{C}$-optimality which would otherwise be the case with isotropic coefficients. As we will see in §3.3.2, such a situation does not occur if rectangular elements are used.

### 3.2.3 Problem 3: heterogeneous, anisotropic and full-tensor $\mathcal{C}$

This test problem is reported in Younes & Fontaine (2008*b*,*a*), Younes et al. (2010). The conductivity field is described by a full-tensor given by

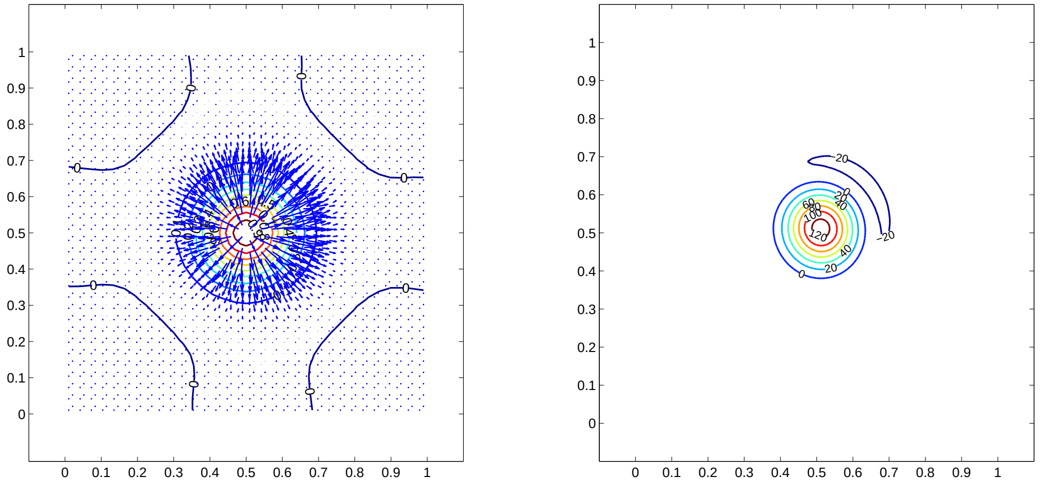
$$\mathcal{C}(\mathbf{x}) = \begin{bmatrix} y^2 + \alpha x^2 & (\alpha - 1)xy \\ (\alpha - 1)xy & x^2 + \alpha y^2 \end{bmatrix}. \tag{3.10}$$

The analytical solution for the potential is given by

$$u(\mathbf{x}) = \exp(-20\pi((x - \frac{1}{2})^2 + (y - \frac{1}{2})^2), \tag{3.11}$$

and the velocity vector is obtained using Darcy's Law $q(\mathbf{x}) = \mathcal{C}(\mathbf{x})\nabla u$. The source term is obtained from $f(\mathbf{x}) = -\nabla \cdot \mathcal{C}(\mathbf{x})\nabla u$.

The MFEM potential and velocity solutions for $h = \frac{1}{32}$ are depicted in Figure 3.3(a) and the source term for $\alpha = 1$ is illustrated in Figure 3.3(b).



(a) Potential $u$ and velocity $\mathbf{q}$ solutions

(b) Source term $f(\mathbf{x})$

Figure 3.3: MFEM solutions and source term for $\alpha = 1$ - Test problem 3

Note that the source term is symmetric with respect to $y = x$ and that the symmetry of the numerical solution improves with mesh refinement.

Error estimates for $\alpha = 1, 10^2, 10^3$ are reported in Tables 3.11, 3.12 and 3.13, respectively. Second order convergence for the potential and first order convergence for the velocities is confirmed also for the full-tensor case. However the magnitude of the errors increases significantly as the order of the anisotropy factor $\alpha$ increases. For $\alpha = 1000$ the error in the potential and velocity solutions is three orders of magnitude larger than for the isotropic case. Hence, for large anisotropy the solution is unphysical and should be considered with care.

A proof of this is given by the minimum and maximum values of the potential solution. This is always positive and ranges from approximately zero close to the boundaries to one at the centre of the domain. For $\alpha = 10^2$ and $\alpha = 10^3$ the minimum and maximum values of the numerical solution are significantly below and above the physical limits of the analytical solution. These unphysical oscillations become less severe for finer meshes, indicating that local mesh refinement could potentially resolve this problem.

Note that spurious oscillations are also present for the isotropic case. This is discordant with results obtained for test isotropic case in test problem 2 (see Table 3.7). Although this is somewhat surprising it largely agrees with results presented by other researchers. Younes & Fontaine (2008*b*) shows that for the same test problem spurious negative oscillations are present on isotropic and anisotropic numerical experiments not only for the MFEM but also for the MPFA method. In the isotropic case the spurious oscillations disappear with mesh refinement, in fact for the case of $h = \frac{1}{512}$ (not shown in Table 3.11) negative oscillations are of the order of $10^{-7}$. Reasons for negative oscillations in the isotropic case are not reported by Younes & Fontaine (2008*b*) and this matter requires further future investigation.

Table 3.11: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 3, $\alpha = 1$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate | $u_{min}$ | $u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $4.29E-03$ | – | $1.25E-01$ | – | $1.25E-01$ | – | $-2.90E-03$ | $9.25E-01$ |
| $\frac{1}{32}$ | $1.08E-03$ | 1.98 | $6.12E-02$ | 1.03 | $6.12E-02$ | 1.03 | $-6.03E-04$ | $9.81E-01$ |
| $\frac{1}{64}$ | $2.72E-04$ | 2.00 | $3.04E-02$ | 1.01 | $3.04E-02$ | 1.01 | $-1.21E-04$ | $9.95E-01$ |
| $\frac{1}{128}$ | $6.80E-05$ | 2.00 | $1.52E-02$ | 1.00 | $1.52E-02$ | 1.00 | $-2.48E-05$ | $9.99E-01$ |
| $\frac{1}{256}$ | $1.70E-05$ | 2.00 | $7.59E-03$ | 1.00 | $7.59E-03$ | 1.00 | $-5.17E-06$ | $1.00E+00$ |

Table 3.12: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 3, $\alpha = 10^2$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate | $u_{min}$ | $u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $5.12E-01$ | – | $1.15E+01$ | – | $1.15E+01$ | – | $-1.56E+00$ | $4.59E+00$ |
| $\frac{1}{32}$ | $1.58E-01$ | 1.70 | $5.43E+00$ | 1.08 | $5.43E+00$ | 1.08 | $-5.20E-01$ | $2.17E+00$ |
| $\frac{1}{64}$ | $4.43E-02$ | 1.84 | $2.54E+00$ | 1.10 | $2.54E+00$ | 1.10 | $-1.43E-01$ | $1.32E+00$ |
| $\frac{1}{128}$ | $1.17E-02$ | 1.92 | $1.23E+00$ | 1.05 | $1.23E+00$ | 1.05 | $-3.74E-02$ | $1.08E+00$ |
| $\frac{1}{256}$ | $2.99E-03$ | 1.97 | $6.07E-01$ | 1.02 | $6.07E-01$ | 1.02 | $-9.42E-03$ | $1.02E+00$ |

Table 3.13: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 3, $\alpha = 10^3$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate | $u_{min}$ | $u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $5.22E+00$ | – | $1.17E+02$ | – | $1.17E+02$ | – | $-1.64E+01$ | $3.82E+01$ |
| $\frac{1}{32}$ | $1.63E+00$ | 1.68 | $5.53E+01$ | 1.08 | $5.53E+01$ | 1.08 | $-6.20E+00$ | $1.31E+01$ |
| $\frac{1}{64}$ | $4.65E-01$ | 1.81 | $2.57E+01$ | 1.10 | $2.57E+01$ | 1.10 | $-1.84E+00$ | $4.30E+00$ |
| $\frac{1}{128}$ | $1.25E-01$ | 1.89 | $1.23E+01$ | 1.06 | $1.23E+01$ | 1.06 | $-4.99E-01$ | $1.85E+00$ |
| $\frac{1}{256}$ | $3.27E-02$ | 1.94 | $6.08E+00$ | 1.02 | $6.08E+00$ | 1.02 | $-1.31E-01$ | $1.21E+00$ |

The computational cost of solving the linear systems of equations using $p-MINRES$ and $PCG$ is reported in Table 3.14. The main results of this table can be summarised as follows:

1. As previously observed for test problem 2, for large degrees of anisotropy the performance of the MINRES solver deteriorates significantly. The larger the value of $\alpha$ the worse it performs;

2. Conversely, $CG$ behaves quite differently for full tensor coefficients. Namely, $CG$ solution timings and iteration counts seems to decrease for increasing $\alpha$. This behaviour is considered to be problem related;

3. For small and medium size meshes, $PCG$ is largely more efficient than $p-MINRES$. However, for finer meshes ($h=\frac{1}{256}$) the cost of implementing the Cholesky factorisation grows significantly. Thus for $\alpha=10^2$, MINRES is more efficient than CG and vice versa for $\alpha=10^3$.

Table 3.14: Iteration count and timings (set-up+solution time) for $p-MINRES$ and $PCG$ - Test problem 3

| | | $p-MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha=10^3$ | 460 | 11.57 | 12 | $0.94+0.16$ |
| | $\alpha=10^2$ | 271 | 6.61 | 21 | $0.95+0.24$ |
| | $\alpha=1$ | 43 | 1.01 | 113 | $1.10+1.17$ |
| $\frac{1}{128}$ | $\alpha=10^3$ | 380 | 49.37 | 14 | $15.56+0.67$ |
| | $\alpha=10^2$ | 316 | 41.25 | 37 | $15.36+1.73$ |
| | $\alpha=1$ | 45 | 5.69 | 220 | $17.09+10.02$ |
| $\frac{1}{256}$ | $\alpha=10^3$ | 474 | 316.87 | 18 | $238.50+4.13$ |
| | $\alpha=10^2$ | 334 | 222.67 | 73 | $238.48+16.31$ |
| | $\alpha=1$ | 45 | 29.40 | 441 | $266.61+97.68$ |

The numerical experiments results using AMG are reported in Table 3.15. These can be summarised as follows:

1. The efficiency of the iterative solvers when used with AMG preconditioners is confirmed also for problems with general full tensor coefficients;

2. In contrast to Table 3.14, the number of $CG$ iterations and solution timings increase with increasing anisotropic coefficient;

3. The $CG$ iteration count is between seven to twenty-one times larger than the reference isotropic case, $\alpha = 1$. This differs significantly from the results recorded for diagonal anisotropic coefficients and indicates that the $AMG$ approximation of the coefficient matrix is not a robust preconditioner for $CG$ when general full-tensor coefficients are used;

4. As for Table 3.14 it is evident that no one solver consistently performs better than the others. Instead, the solvers' performance depends on the size of the mesh and the degree of anisotropy. Thus $p-MINRES_{AMG}$ performs better for fine meshes, $h = \frac{1}{256}$, and $PCG_{AMG}$ performs better for medium to small size meshes.

Table 3.15: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 3

| | | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^3$ | 475 | $1.58 + 4.50$ | 195 | $2.22 + 1.94$ |
| | $\alpha = 10^2$ | 285 | $1.60 + 2.84$ | 64 | $2.18 + 0.66$ |
| | $\alpha = 1$ | 50 | $1.27 + 0.46$ | 8 | $1.97 + 0.12$ |
| $\frac{1}{128}$ | $\alpha = 10^3$ | 415 | $9.63 + 18.63$ | 192 | $14.73 + 8.84$ |
| | $\alpha = 10^2$ | 345 | $9.51 + 15.32$ | 65 | $15.03 + 2.93$ |
| | $\alpha = 1$ | 52 | $8.14 + 2.19$ | 9 | $14.39 + 0.43$ |
| $\frac{1}{256}$ | $\alpha = 10^3$ | 546 | $102.33 + 129.59$ | 192 | $249.91 + 49.38$ |
| | $\alpha = 10^2$ | 383 | $101.38 + 91.70$ | 66 | $252.64 + 17.03$ |
| | $\alpha = 1$ | 54 | $111.51 + 11.36$ | 9 | $271.42 + 2.14$ |

### 3.2.4 Problem 4: discontinuous, anisotropic and full-tensor $\mathcal{C}$

This test problem was originally presented in Crumpton et al. (1995). Using this example we intend to assess the efficiency and accuracy of MFEM for cases in which the conductivity coefficient is strongly discontinuous. This is a situation which is very often encountered in applications and therefore of significant importance for this work.

Define $D = [-1, 1]^2$ and $\mathcal{C}$ is given by

$$\mathcal{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ for } x < 0, \qquad \mathcal{C} = \alpha \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \text{ for } x > 0. \tag{3.12}$$

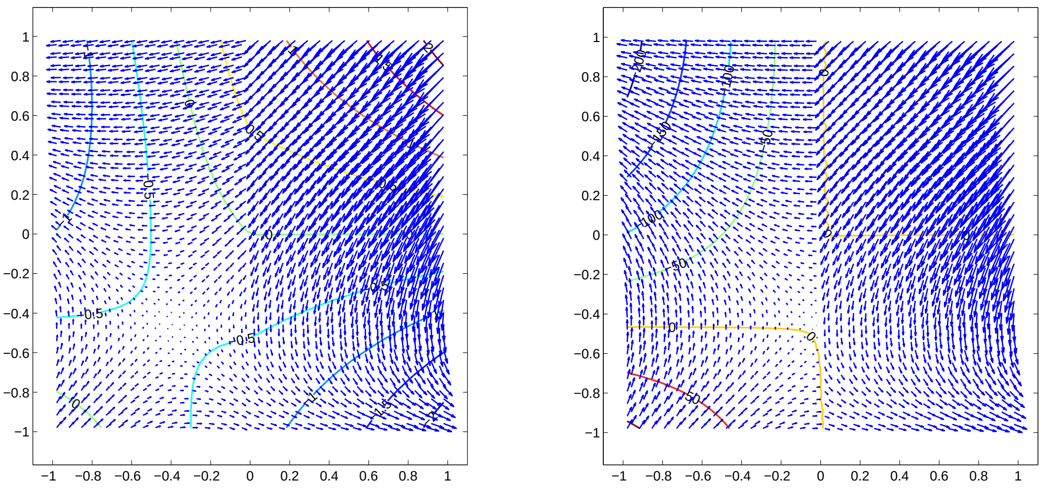
The parameter $\alpha$ controls the strength of the discontinuity at $x = 0$. The exact solution for this test problem is given by

$$u(\mathbf{x}) = \begin{cases} (2\sin(y) + \cos(y))\alpha x + \sin(y) & \text{for } x < 0, \\ \exp(x)\sin(y) & \text{for } x > 0. \end{cases} \tag{3.13}$$

The MFEM solutions for $\alpha = 1$ and $\alpha = 100$, for $h = \frac{1}{32}$ are illustrated in Figure 3.4.

Error estimates for $\alpha = 1$ are reported in Table 3.16. For this test problem we observe the loss of one order of magnitude in the rate of convergence for the potential solution. However, the velocity solution retains the characteristic first order convergence rate which was recorded also for the other test problems. The error for the potential solution is located at the discontinuity and it vanishes as $h$ is progressively refined. Local mesh refinement at the location of the discontinuity should enhance the rate of convergence in the potential solution.

Tables 3.17 and 3.18 report discrete error estimates for $\alpha = 10^1$ and $\alpha = 10^2$. Interestingly, the magnitude of the errors in the potential solution are of the same order as those reported for $\alpha = 1$. In contrast, the velocity errors are one and two



(a) $\alpha = 1$

(b) $\alpha = 10^2$

Figure 3.4: MFEM solutions for $\alpha = 1, 10^2$ - Test problem 4

orders larger, respectively. Noticeably, the potential convergence rate is slightly lower than one for $\alpha = 10$ and $h = \frac{1}{256}$ and it approaches $\mathcal{O}(h^{\frac{3}{2}})$ for $\alpha = 100$ and $h = \frac{1}{256}$.

Table 3.16: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 4, $\alpha = 1$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $8.06E-03$ | – | $1.77E-01$ | – | $1.71E-01$ | – |
| $\frac{1}{32}$ | $3.35E-03$ | 1.27 | $8.91E-02$ | 0.99 | $8.55E-02$ | 1.00 |
| $\frac{1}{64}$ | $1.63E-03$ | 1.04 | $4.46E-02$ | 1.00 | $4.27E-02$ | 1.00 |
| $\frac{1}{128}$ | $8.27E-04$ | 0.98 | $2.23E-02$ | 1.00 | $2.13E-02$ | 1.00 |
| $\frac{1}{256}$ | $4.19E-04$ | 0.98 | $1.12E-02$ | 1.00 | $1.07E-02$ | 1.00 |

Table 3.17: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 4, $\alpha = 10^1$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.34E-02$ | – | $1.77E+00$ | – | $1.73E+00$ | – |
| $\frac{1}{32}$ | $4.62E-03$ | 1.54 | $8.92E-01$ | 0.99 | $8.67E-01$ | 1.00 |
| $\frac{1}{64}$ | $2.22E-03$ | 1.06 | $4.47E-01$ | 1.00 | $4.33E-01$ | 1.00 |
| $\frac{1}{128}$ | $1.17E-03$ | 0.93 | $2.24E-01$ | 1.00 | $2.17E-01$ | 1.00 |
| $\frac{1}{256}$ | $6.05E-04$ | 0.95 | $1.12E-01$ | 1.00 | $1.08E-01$ | 1.00 |

Solver performances for test problem 4 are reported in Tables 3.19 and 3.20. The

Table 3.18: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 4, $\alpha = 10^2$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.17E-01$ | – | $1.77E+01$ | – | $1.74E+01$ | – |
| $\frac{1}{32}$ | $2.90E-02$ | 2.01 | $8.94E+00$ | 0.99 | $8.69E+00$ | 1.00 |
| $\frac{1}{64}$ | $7.14E-03$ | 2.02 | $4.48E+00$ | 1.00 | $4.34E+00$ | 1.00 |
| $\frac{1}{128}$ | $1.89E-03$ | 1.92 | $2.24E+00$ | 1.00 | $2.17E+00$ | 1.00 |
| $\frac{1}{256}$ | $6.60E-04$ | 1.52 | $1.12E+00$ | 1.00 | $1.09E+00$ | 1.00 |

results reported in these two tables can be summarised as follows:

1. MINRES iteration count for problems with discontinuities is larger (between 30% to 40%) than for continuous problems. The same behaviour is observed for the exact and approximated versions of preconditioner (2.41);

2. It appears that the exact version of $p - MINRES$ is by far the most efficient solver for problems with discontinuities. For all other methods considered the CPU time required to either implement the factorisation or construct the coarse grids significantly penalises the performance of the solvers;

3. For all methods the order (governed by $\alpha$) of the discontinuity has virtually no effect on the solvers performance. It appears that for larger $\alpha$, i.e. sharper variation in the conductivity at the discontinuity, the number of iterations is smaller than for smaller $\alpha$, i.e. more homogeneous conditions at the discontinuity;

### 3.2.5 Problem 5: distorted triangular mesh

In this section the behaviour of the numerical methods on distorted meshes is assessed. Although the mesh is distorted the finite element connectivity is regular, i.e. any node has the same number of neighboring nodes. Experiments on irregular connectivity are not reported in this thesis.

Table 3.19: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 4

| | | $p - MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 65 | 1.23 | 85 | $0.97 + 0.87$ |
| | $\alpha = 10^1$ | 68 | 1.31 | 84 | $0.95 + 0.85$ |
| | $\alpha = 1$ | 68 | 1.32 | 83 | $0.94 + 0.83$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 65 | 8.69 | 165 | $15.48 + 7.47$ |
| | $\alpha = 10^1$ | 67 | 8.77 | 165 | $15.27 + 7.62$ |
| | $\alpha = 1$ | 68 | 9.26 | 162 | $15.33 + 7.27$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 64 | 44.08 | 325 | $240.71 + 73.00$ |
| | $\alpha = 10^1$ | 67 | 48.45 | 323 | $238.17 + 73.01$ |
| | $\alpha = 1$ | 68 | 45.95 | 318 | $245.11 + 71.28$ |

Table 3.20: Iteration count and timings (set-up+solution time) for $p - MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 4

| | | $p - MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 78 | $1.33 + 0.75$ | 12 | $2.12 + 0.16$ |
| | $\alpha = 10^1$ | 82 | $1.38 + 0.78$ | 12 | $2.08 + 0.16$ |
| | $\alpha = 1$ | 83 | $1.35 + 0.74$ | 12 | $2.12 + 0.16$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 79 | $8.15 + 3.21$ | 12 | $14.63 + 0.64$ |
| | $\alpha = 10^1$ | 83 | $8.08 + 3.36$ | 12 | $14.76 + 0.58$ |
| | $\alpha = 1$ | 84 | $8.04 + 3.44$ | 12 | $15.03 + 0.61$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 78 | $102.86 + 20.72$ | 13 | $278.41 + 3.70$ |
| | $\alpha = 10^1$ | 85 | $108.85 + 21.10$ | 13 | $252.37 + 3.28$ |
| | $\alpha = 1$ | 85 | $114.45 + 21.39$ | 12 | $260.69 + 2.93$ |

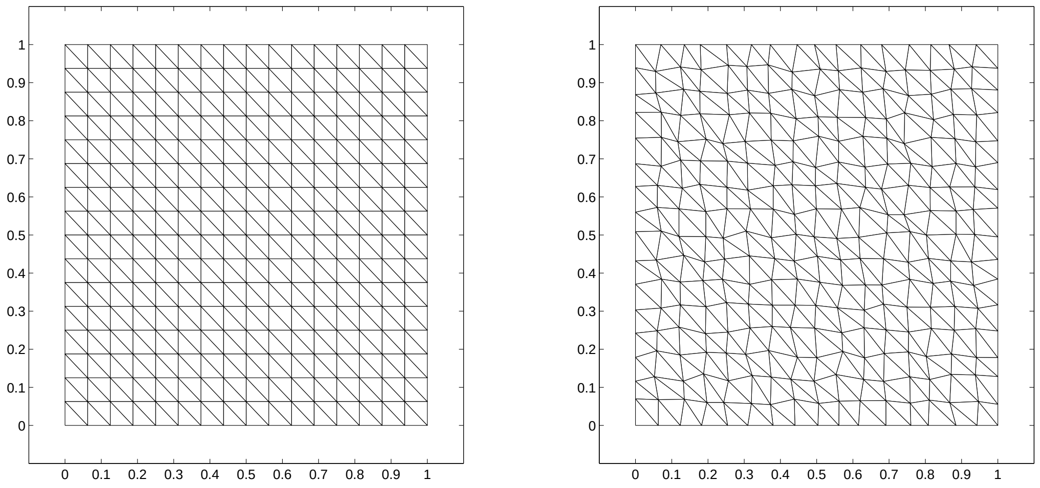
The test problem is taken from Arnold et al. (2005). The analytical solution on the unit square domain is $u = x(1-x)y(1-y)$. The conductivity coefficient is a unit scalar. Therefore, (2.1) simplifies to the Poisson's equation in this case.

The distorted mesh is created perturbing the node coordinates of the original structured mesh according to

$$\mathbf{x}_{unst} = \mathbf{x}_{st} + zh^{\alpha}$$

where $z$ is a uniformly distributed random number in the range $[-0.5, 0.5]$ and $\alpha$ regulates the order of the perturbation. Distorted meshes are created at each discretisation level.

An example of structured and unstructured meshes used for this test problem is given in Figure 3.5. For the experiments herein considered $\alpha = 1.2$.



(a) Structured mesh

(b) Distorted mesh

Figure 3.5: Structured and perturbed triangular finite element mesh for $h = \frac{1}{16}$ - Test problem 5

Discrete error estimates for the structured and unstructured cases are reported in Table 3.22. It appears that the magnitude of the errors and the convergence rate are not affected by the irregular meshing. Hence the potential converges with rate $\mathcal{O}(h^2)$ and the velocities with rate $\mathcal{O}(h)$. It is clear that the mixed method is also suitable for accurate approximations on distorted meshes.

The performance of the solvers is reported in Tables 3.22 and 3.23. The results reported in the tables can be summarised as follow:

1. Both versions of preconditioned MINRES are $h$-optimal. For the unstructured case the iteration count is slightly larger and some small variations with $h$ are recorded.

2. CG using the incomplete Cholesky factorisation of the coefficient matrix is not

Table 3.21: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 5

| *Structured Meshes* | | | | | | |
|---|---|---|---|---|---|---|
| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
| $\frac{1}{16}$ | $6.32E-05$ | – | $4.08E-03$ | – | $4.08E-03$ | – |
| $\frac{1}{32}$ | $1.59E-05$ | 1.99 | $2.05E-03$ | 0.99 | $2.05E-03$ | 0.99 |
| $\frac{1}{64}$ | $3.98E-06$ | 2.00 | $1.03E-03$ | 1.00 | $1.03E-03$ | 1.00 |
| $\frac{1}{128}$ | $9.97E-07$ | 2.00 | $5.13E-04$ | 1.00 | $5.13E-04$ | 1.00 |
| $\frac{1}{256}$ | $2.49E-07$ | 2.00 | $2.57E-04$ | 1.00 | $2.57E-04$ | 1.00 |
| *Unstructured Meshes* | | | | | | |
| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
| $\frac{1}{16}$ | $6.91E-05$ | – | $4.51E-03$ | – | $4.90E-03$ | – |
| $\frac{1}{32}$ | $1.78E-05$ | 1.96 | $2.41E-03$ | 0.90 | $2.37E-03$ | 1.05 |
| $\frac{1}{64}$ | $4.30E-06$ | 2.05 | $1.16E-03$ | 1.06 | $1.16E-03$ | 1.03 |
| $\frac{1}{128}$ | $1.06E-06$ | 2.02 | $5.66E-04$ | 1.03 | $5.65E-04$ | 1.04 |
| $\frac{1}{256}$ | $2.60E-07$ | 2.02 | $2.77E-04$ | 1.03 | $2.77E-04$ | 1.03 |

$h$-optimal. Also for this method a larger iteration count is recorded for unstructured meshes.

3. The *AMG* version of CG is $h$-optimal. As for the other test problems the efficient performance of CG is penalised by the large CPU cost of creating the *AMG* grids;

Table 3.22: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 5

| *Structured Meshes* | | | | |
|---|---|---|---|---|
| | $p - MINRES$ | | $PCG$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 43 | 0.82 | 91 | $0.96 + 0.95$ |
| $\frac{1}{128}$ | 43 | 5.43 | 164 | $15.01 + 7.83$ |
| $\frac{1}{256}$ | 43 | 28.78 | 310 | $243.92 + 70.11$ |
| *Unstructured Meshes* | | | | |
| | $p - MINRES$ | | $PCG$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 52 | 0.99 | 97 | $0.97 + 1.04$ |
| $\frac{1}{128}$ | 51 | 6.56 | 190 | $15.25 + 9.24$ |
| $\frac{1}{256}$ | 49 | 32.68 | 369 | $237.46 + 82.39$ |

Table 3.23: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 5

| | *Structured Meshes* | | | |
|---|---|---|---|---|
| | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 48 | $1.28 + 0.45$ | 9 | $1.97 + 0.15$ |
| $\frac{1}{128}$ | 48 | $8.01 + 2.02$ | 9 | $13.65 + 0.46$ |
| $\frac{1}{256}$ | 48 | $112.57 + 10.81$ | 10 | $224.68 + 2.63$ |
| | *Unstructured Meshes* | | | |
| | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 63 | $1.62 + 0.66$ | 16 | $2.56 + 0.30$ |
| $\frac{1}{128}$ | 61 | $9.05 + 3.08$ | 16 | $17.39 + 0.85$ |
| $\frac{1}{256}$ | 63 | $112.91 + 15.00$ | 14 | $277.21 + 3.47$ |

## 3.3 Numerical experiments on rectangular meshes

In this section the numerical experiments previously carried out on triangular meshes are performed on rectangular meshes. It is worthwhile to anticipate that all findings highlighted in the previous sections are also valid for rectangular meshes. However, for some test problems, there are some major differences with respect to the triangular case.

### 3.3.1 Problem 1: heterogeneous, isotropic and diagonal $\mathcal{C}$

Tables 3.24, 3.25 and 3.26 report discrete error estimates for $\epsilon = 0.9, 0.99, 0.999$, respectively. For the case of small heterogeneity, i.e. $\epsilon = 0.9$, the solutions converge with order larger than second. In fact, the convergence rate for the potential is $\mathcal{O}(h^{2.08})$ and for the $x$ and $y$ components of the velocity field are $\mathcal{O}(h^{2.16})$ and $\mathcal{O}(h^{2.21})$, respectively. This is significantly different from the convergence rates observed on triangular meshes, whereby first order convergence was recorded for the velocity solution (see §3.2.1).

Furthermore, for the same level of discretisation, the magnitude of the error potential for the rectangular case is lower than the triangular case. For the velocity solution this is two orders of magnitude lower.

For the case of moderate heterogeneity, i.e. $\epsilon = 0.99$, larger convergence rates are recorded for the potential solution, $\mathcal{O}(h^{2.28})$. However the velocities components converge at rates $\mathcal{O}(h^{1.02})$ and $\mathcal{O}(h^{1.46})$, respectively. Although these rates are lower than for the case of $\epsilon = 0.9$, these are significantly better than the triangular case.

For the case $\epsilon = 0.999$ the convergence rates and the magnitude of the error are comparable to those recorded for the triangular case.

Table 3.24: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 1, $\epsilon = 0.9$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $2.94E-03$ | – | $1.66E-01$ | – | $8.34E-02$ | – |
| $\frac{1}{32}$ | $6.10E-04$ | 2.27 | $3.45E-02$ | 2.27 | $9.47E-03$ | 3.14 |
| $\frac{1}{64}$ | $1.49E-04$ | 2.03 | $6.87E-03$ | 2.33 | $2.96E-03$ | 1.68 |
| $\frac{1}{128}$ | $3.72E-05$ | 2.00 | $1.69E-03$ | 2.03 | $7.29E-04$ | 2.02 |
| $\frac{1}{256}$ | $9.31E-06$ | 2.00 | $4.20E-04$ | 2.01 | $1.81E-04$ | 2.01 |

Table 3.25: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 1, $\epsilon = 0.99$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $8.57E-03$ | – | $6.42E-01$ | – | $1.56E+00$ | – |
| $\frac{1}{32}$ | $2.43E-03$ | 1.82 | $4.82E-01$ | 0.41 | $1.48E+00$ | 0.07 |
| $\frac{1}{64}$ | $5.88E-04$ | 2.05 | $3.14E-01$ | 0.62 | $1.04E+00$ | 0.50 |
| $\frac{1}{128}$ | $1.06E-04$ | 2.47 | $1.47E-01$ | 1.10 | $3.64E-01$ | 1.52 |
| $\frac{1}{256}$ | $1.54E-05$ | 2.78 | $3.77E-02$ | 1.96 | $2.72E-02$ | 3.74 |

Table 3.26: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 1, $\epsilon = 0.999$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $9.78E-03$ | – | $7.11E-01$ | – | $1.87E+00$ | – |
| $\frac{1}{32}$ | $3.18E-03$ | 1.62 | $5.98E-01$ | 0.25 | $2.18E+00$ | $< 0$ |
| $\frac{1}{64}$ | $1.02E-03$ | 1.64 | $5.00E-01$ | 0.26 | $2.48E+00$ | $< 0$ |
| $\frac{1}{128}$ | $3.22E-04$ | 1.67 | $4.07E-01$ | 0.29 | $2.69E+00$ | $< 0$ |
| $\frac{1}{256}$ | $9.50E-05$ | 1.76 | $3.15E-01$ | 0.37 | $2.67E+00$ | 0.01 |

The solvers performance is recorded in Table 3.27. The same findings summarised in §3.2.1 for triangular meshes also apply to rectangular meshes. In addition to those it should be noted that:

1. Solvers' CPU timings for the rectangular case are significantly lower than the triangular case. This is obviously associated with the smaller size of the coefficient matrix in the former case. For the same reason the cost of implementing the Cholesky factorisation is considerably lower;

2. $p - MINRES$ iteration count for the rectangular case is comparable to the triangular case. Although a slightly larger variability is recorded, $h$-optimality and $\mathcal{C}$-optimality is preserved;

3. In contrast to the MINRES solver, the CG iteration count for the rectangular case is significantly smaller than the triangular case.

Table 3.27: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 1

| | | $p - MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\epsilon$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\epsilon = 0.999$ | 45 | 0.65 | 88 | $0.55 + 0.62$ |
| | $\epsilon = 0.99$ | 46 | 0.62 | 89 | $0.57 + 0.62$ |
| | $\epsilon = 0.9$ | 44 | 0.67 | 97 | $0.56 + 0.70$ |
| $\frac{1}{128}$ | $\epsilon = 0.999$ | 45 | 3.35 | 170 | $8.72 + 5.45$ |
| | $\epsilon = 0.99$ | 46 | 3.31 | 173 | $8.66 + 5.48$ |
| | $\epsilon = 0.9$ | 39 | 2.78 | 189 | $8.77 + 5.98$ |
| $\frac{1}{256}$ | $\epsilon = 0.999$ | 44 | 15.72 | 336 | $132.36 + 50.83$ |
| | $\epsilon = 0.99$ | 45 | 16.63 | 337 | $136.88 + 52.07$ |
| | $\epsilon = 0.9$ | 34 | 12.29 | 371 | $135.56 + 57.25$ |

The results for the AMG experiments are reported in Table 3.28. The considerations highlighted in §3.2.1 regarding Table 3.5 are equally valid for rectangular meshes. Additionally we note that:

1. The CPU cost of constructing the AMG grids is significantly lower than the triangular case. Specifically, it is four times smaller for the Schur complement and three times smaller for the MHFEM coefficient matrix;

2. Given the smaller size of the system of equations MINRES and CG CPU cost are significantly lower than the triangular case;

3. For isotropic coefficients, the AMG versions of MINRES and CG are efficient and robust solvers. However, their overall performance is penalised by the CPU cost of creating the AMG grids which is not negligible also for rectangular meshes.

Table 3.28: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 1

| | | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\epsilon$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\epsilon = 0.999$ | 57 | $0.71 + 0.33$ | 12 | $0.19 + 0.12$ |
| | $\epsilon = 0.99$ | 56 | $0.71 + 0.34$ | 13 | $0.19 + 0.13$ |
| | $\epsilon = 0.9$ | 55 | $0.66 + 0.30$ | 12 | $0.19 + 0.13$ |
| $\frac{1}{128}$ | $\epsilon = 0.999$ | 57 | $3.18 + 1.05$ | 13 | $6.55 + 0.43$ |
| | $\epsilon = 0.99$ | 57 | $3.24 + 1.06$ | 12 | $6.52 + 0.39$ |
| | $\epsilon = 0.9$ | 57 | $3.21 + 1.11$ | 13 | $6.60 + 0.42$ |
| $\frac{1}{256}$ | $\epsilon = 0.999$ | 59 | $25.11 + 6.67$ | 13 | $99.00 + 2.09$ |
| | $\epsilon = 0.99$ | 61 | $25.21 + 7.11$ | 13 | $99.13 + 2.09$ |
| | $\epsilon = 0.9$ | 57 | $25.25 + 6.62$ | 13 | $98.75 + 2.10$ |

### 3.3.2 Problem 2: heterogeneous, anisotropic and diagonal $\mathcal{C}$

The settings for this test problem are described in §3.2.2. The error estimates on rectangular meshes are reported in Tables 3.29, 3.30 and 3.31 for $\alpha = 10^{-2}, 1, 10^{2}$. Second order convergence $\mathcal{O}(h^2)$ is recorded for the potential and velocity solutions. Note that the same convergence rates are obtained for all values of the anisotropic coefficient, $\alpha$. Furthermore the errors are approximately of the same order of magnitude.

As explained for the triangular case, the potential solution for this test problem is always positive and specifically it is 0 at the boundaries and 0.0625 at the center of the domain, so that $0 < u(\mathbf{x}) < 0.0625, \forall \mathbf{x} \in D$. On triangular meshes and for $\alpha \neq 1$

(see Table 3.28), the numerical solution presents unphysical negative oscillations. According to results shown in Tables 3.29, 3.30 and 3.31, the potential solution does not exhibit this erroneous behaviour on rectangular meshes.

Table 3.29: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 2, $\alpha = 10^{-2}$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate | $u_{min}$ |
|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.00E-04$ | – | $1.17E-04$ | – | $2.40E-04$ | – | $8.28E-04$ |
| $\frac{1}{32}$ | $2.55E-05$ | 1.97 | $2.96E-05$ | 1.99 | $6.03E-05$ | 1.99 | $2.02E-04$ |
| $\frac{1}{64}$ | $6.39E-06$ | 1.99 | $7.42E-06$ | 2.00 | $1.51E-05$ | 2.00 | $4.95E-05$ |
| $\frac{1}{128}$ | $1.60E-06$ | 2.00 | $1.86E-06$ | 2.00 | $3.78E-06$ | 2.00 | $1.22E-05$ |
| $\frac{1}{256}$ | $4.00E-07$ | 2.00 | $4.64E-07$ | 2.00 | $9.45E-07$ | 2.00 | $3.03E-06$ |

Table 3.30: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 2, $\alpha = 1$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate | $u_{min}$ |
|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.46E-04$ | – | $2.98E-04$ | – | $2.98E-04$ | – | $7.69E-04$ |
| $\frac{1}{32}$ | $3.70E-05$ | 1.98 | $7.50E-05$ | 1.99 | $7.50E-05$ | 1.99 | $1.82E-04$ |
| $\frac{1}{64}$ | $9.29E-06$ | 1.99 | $1.88E-05$ | 2.00 | $1.88E-05$ | 2.00 | $4.37E-05$ |
| $\frac{1}{128}$ | $2.32E-06$ | 2.00 | $4.70E-06$ | 2.00 | $4.70E-06$ | 2.00 | $1.06E-05$ |
| $\frac{1}{256}$ | $5.81E-07$ | 2.00 | $1.17E-06$ | 2.00 | $1.17E-06$ | 2.00 | $2.60E-06$ |

Table 3.31: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 2, $\alpha = 10^{2}$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate | $u_{min}$ |
|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $6.08E-04$ | – | $4.91E-02$ | – | $1.07E-03$ | – | $5.41E-04$ |
| $\frac{1}{32}$ | $1.56E-04$ | 1.97 | $1.24E-02$ | 1.98 | $2.72E-04$ | 1.98 | $1.17E-04$ |
| $\frac{1}{64}$ | $3.88E-05$ | 2.00 | $3.11E-03$ | 2.00 | $6.82E-05$ | 2.00 | $2.58E-05$ |
| $\frac{1}{128}$ | $9.66E-06$ | 2.00 | $7.79E-04$ | 2.00 | $1.71E-05$ | 2.00 | $5.86E-06$ |
| $\frac{1}{256}$ | $2.41E-06$ | 2.00 | $1.95E-04$ | 2.00 | $4.27E-06$ | 2.00 | $1.37E-06$ |

The solvers' computational performance for problem 2 on rectangular meshes is presented in Table 3.32 and can be summarised as follows:

1. As opposed to the experiments carried out on triangular meshes (see Table 3.9), preconditioned MINRES is $\mathcal{C}$-optimal when the conductivity coefficient is diagonal and anisotropic;

2. MINRES performance (in terms of $N_{it}$ and $t_{CPU}$) is completely independent of the degree of anisotropy;

3. CG performance is comparable to the one reported for triangular meshes, i.e. it is neither $h$ nor $\mathcal{C}$ optimal;

Table 3.32: Iteration count and timings (set-up+solution time) for $p-MINRES$ and $PCG$ - Test problem 2

| | | $p-MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 38 | 0.65 | 68 | $0.57 + 0.49$ |
| | $\alpha = 1$ | 36 | 0.58 | 81 | $0.59 + 0.57$ |
| | $\alpha = 10^{-2}$ | 37 | 0.58 | 83 | $0.56 + 0.59$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 33 | 2.54 | 136 | $8.82 + 4.35$ |
| | $\alpha = 1$ | 33 | 2.53 | 158 | $8.66 + 4.99$ |
| | $\alpha = 10^{-2}$ | 33 | 2.56 | 170 | $8.65 + 5.35$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 29 | 10.51 | 274 | $133.96 + 42.05$ |
| | $\alpha = 1$ | 29 | 10.51 | 311 | $135.29 + 47.04$ |
| | $\alpha = 10^{-2}$ | 30 | 10.78 | 353 | $139.34 + 54.00$ |

The results for the AMG numerical experiments are reported in Table 3.33. The optimality of preconditioned MINRES, previously discussed, is also valid when the Schur complement is approximated by one V-cycle of black-box AMG. In addition to this, it is evident from Table 3.33 that:

1. In contrast to the experiments on triangular meshes, the number of MINRES iterations is approximately constant for $\alpha \neq 1$. Not surprisingly, for the isotropic case, $N_{it}$ is generally lower;

2. For $\alpha \neq 1$, the number of CG iterations varies considerably. This is not the case for the experiments on triangular meshes (see Table 3.10). Reasons for the difference in performance between triangular and rectangular meshes are given below.

Table 3.33: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 2

| | | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 52 | $0.77 + 0.35$ | 12 | $1.27 + 0.13$ |
| | $\alpha = 1$ | 47 | $0.67 + 0.28$ | 15 | $1.23 + 0.14$ |
| | $\alpha = 10^{-2}$ | 54 | $0.73 + 0.33$ | 19 | $1.24 + 0.17$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 56 | $3.73 + 1.23$ | 13 | $6.15 + 0.54$ |
| | $\alpha = 1$ | 46 | $3.16 + 0.87$ | 15 | $6.66 + 0.49$ |
| | $\alpha = 10^{-2}$ | 53 | $3.36 + 1.04$ | 20 | $6.62 + 0.62$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 59 | $28.74 + 7.25$ | 14 | $87.01 + 2.62$ |
| | $\alpha = 1$ | 46 | $25.61 + 5.34$ | 16 | $100.19 + 2.74$ |
| | $\alpha = 10^{-2}$ | 56 | $26.48 + 6.30$ | 21 | $95.46 + 3.37$ |

As pointed out in Table 3.33, the number of CG iterations varies considerably for $\alpha \neq 1$. This is due to the fact that the coefficient matrix $D$ is not an $M$-matrix for anisotropic diagonal tensors and rectangular meshes (Powell 2003). The black-box AMG code used in this work is set up to work with $M$-matrices. When this condition is violated the performance of black-box AMG can deteriorate significantly.

For triangular elements with diagonal-anisotropic coefficients, the Lagrange multiplier system $D$ is always an $M$-matrix, hence AMG behaviour is not erratic and the number of CG iterations tends to vary only slightly for $\alpha \neq 1$ (see Table 3.10). Furthermore, as proved by (Powell 2003), the Schur complement $(BAB^T)$, which is used as preconditioner for MINRES, is always an $M$-matrix, hence the optimal performance of the AMG code is always guaranteed.

Preconditioned MINRES is $\mathcal{C}$-optimal for diagonal anisotropic conductivity coefficients on rectangular meshes due to the structure of the element stiffness matrix $A_K$. Powell (2003), Powell & Silvester (2003) showed, in fact, that $A_K$ for rectangular elements has diagonal blocks and each block is scaled by a different entry of the coefficient $\mathcal{C}$. This is very different from triangular elements where every row of $A_K$ is scaled by all entries of the coefficient $\mathcal{C}$.

### 3.3.3 Problem 3: heterogeneous, anisotropic and full-tensor $\mathcal{C}$

$L^2(D^h)$ error estimates for test problem 3 on rectangular meshes are reported in Tables 3.34, 3.35 and 3.36 for various values of $\alpha$.

Second order convergence rates for the potential and velocity solutions are also confirmed for problems with full-tensor, anisotropic coefficients. As for triangular meshes, the magnitude of the discrete errors increases with larger anisotropic coefficients.

As for the triangular case negative oscillations for the potential solution are also recorded for rectangular elements. Younes & Fontaine (2008*a*) reported numerical experiments using the MFEM and MPFA for the same test problem reported in this section. The authors show numerical results which are largely consistent with the results reported in Tables 3.34, 3.35 and 3.36, i.e spurious negative oscillations are present not only for the anisotropic case but also for the isotropic case. For the isotropic case the spurious oscillations disappear with mesh refinement, in fact for the case of $h = \frac{1}{512}$ (not shown in Table 3.34) negative oscillations are of the order of $10^{-8}$. Reasons for negative oscillations in the isotropic case are not reported by Younes & Fontaine (2008*a*) and this matter requires further future investigation.

The solvers' performance for test problem 3 on rectangular meshes is reported in Table 3.37. The main findings of this table can be summarised as follows:

1. As for triangular elements, the performance of MINRES deteriorates significantly for large values of $\alpha$;

2. Conversely, CG behaves quite differently for full tensor coefficients since the CPU cost seems to decrease with increasing $\alpha$. Similar results were obtained for triangular meshes;

3. For $\alpha \neq 1$ CG outperforms MINRES for all discretisation levels;

Table 3.34: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 3, $\alpha = 1$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate | $u_{min}$ | $u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.25E-02$ | – | $9.29E-02$ | – | $9.29E-02$ | – | $-9.29E-04$ | $8.10E-01$ |
| $\frac{1}{32}$ | $3.35E-03$ | 1.91 | $2.43E-02$ | 1.93 | $2.43E-02$ | 1.93 | $-2.34E-04$ | $9.44E-01$ |
| $\frac{1}{64}$ | $8.51E-04$ | 1.98 | $6.15E-03$ | 1.98 | $6.15E-03$ | 1.98 | $-3.71E-05$ | $9.85E-01$ |
| $\frac{1}{128}$ | $2.14E-04$ | 1.99 | $1.54E-03$ | 2.00 | $1.54E-03$ | 2.00 | $-5.09E-06$ | $9.96E-01$ |
| $\frac{1}{256}$ | $5.35E-05$ | 2.00 | $3.86E-04$ | 2.00 | $3.86E-04$ | 2.00 | $-4.21E-07$ | $9.99E-01$ |

Table 3.35: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 3, $\alpha = 10^2$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate | $u_{min}$ | $u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.73E-01$ | – | $7.22E+00$ | – | $7.22E+00$ | – | $-4.92E-01$ | $2.03E+00$ |
| $\frac{1}{32}$ | $4.83E-02$ | 1.84 | $1.96E+00$ | 1.88 | $1.96E+00$ | 1.88 | $-1.20E-01$ | $1.36E+00$ |
| $\frac{1}{64}$ | $1.26E-02$ | 1.94 | $5.04E-01$ | 1.96 | $5.04E-01$ | 1.96 | $-2.71E-02$ | $1.10E+00$ |
| $\frac{1}{128}$ | $3.18E-03$ | 1.98 | $1.27E-01$ | 1.98 | $1.27E-01$ | 1.98 | $-6.62E-03$ | $1.02E+00$ |
| $\frac{1}{256}$ | $7.98E-04$ | 1.99 | $3.20E-02$ | 1.99 | $3.20E-02$ | 1.99 | $-1.60E-03$ | $1.01E+00$ |

Table 3.36: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 3, $\alpha = 10^3$

| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate | $u_{min}$ | $u_{max}$ |
|---|---|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.88E+00$ | – | $7.30E+01$ | – | $7.30E+01$ | – | $-5.82E+00$ | $1.34E+01$ |
| $\frac{1}{32}$ | $5.31E-01$ | 1.83 | $2.00E+01$ | 1.87 | $2.00E+01$ | 1.87 | $-2.02E+00$ | $5.24E+00$ |
| $\frac{1}{64}$ | $1.40E-01$ | 1.93 | $5.20E+00$ | 1.94 | $5.20E+00$ | 1.94 | $-4.95E-01$ | $2.13E+00$ |
| $\frac{1}{128}$ | $3.57E-02$ | 1.97 | $1.34E+00$ | 1.96 | $1.34E+00$ | 1.96 | $-1.06E-01$ | $1.28E+00$ |
| $\frac{1}{256}$ | $9.01E-03$ | 1.99 | $3.40E-01$ | 1.97 | $3.40E-01$ | 1.97 | $-2.57E-02$ | $1.07E+00$ |

Note that for $\alpha = 1$, the conductivity coefficient is equivalent to that of test problem 2. The only difference between the two problems is associated with the right-hand side of the PDE. In such circumstances it is normally expected for MINRES to converge with approximately the same number of iterations. However, from Table 3.37 it is evident that the number of iterations required to solve problem 3 on rectangular meshes is significantly lower than problem 2. This behaviour is not observed for triangular meshes.

Table 3.37: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 3

| | | $p - MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^3$ | 351 | 5.07 | 12 | $0.57 + 0.13$ |
| | $\alpha = 10^2$ | 254 | 3.81 | 22 | $0.55 + 0.18$ |
| | $\alpha = 1$ | 22 | 0.33 | 79 | $0.54 + 0.55$ |
| $\frac{1}{128}$ | $\alpha = 10^3$ | 466 | 38.08 | 14 | $8.63 + 0.50$ |
| | $\alpha = 10^2$ | 283 | 21.82 | 40 | $8.64 + 1.29$ |
| | $\alpha = 1$ | 18 | 1.37 | 154 | $8.58 + 4.77$ |
| $\frac{1}{256}$ | $\alpha = 10^3$ | 554 | 204.61 | 20 | $133.07 + 3.06$ |
| | $\alpha = 10^2$ | 300 | 111.65 | 77 | $134.12 + 11.60$ |
| | $\alpha = 1$ | 13 | 4.76 | 306 | $134.98 + 45.43$ |

Results for the AMG numerical experiments are reported in Table 3.38. The most important observations for this table can be summarised as follow:

1. The MINRES iteration count grows rapidly with increasing $\alpha$, hence the solution timings are quite large. However, given that the CPU cost of constructing the coarse grids for the AMG approximation is quite cheap on rectangular meshes, $p - MINRES_{AMG}$ is the better performing solver among all considered;

2. The performance of CG is significantly different from the one reported for triangular elements. On triangular meshes, although $\mathcal{C}$-optimality is not obtained, CG is $h$-optimal. On rectangular meshes neither $\mathcal{C}$ nor $h$ optimality is established. This aspect is associated with the violation of the $M$-matrix condition

for problems with full-tensor coefficients;

Table 3.38: Iteration count and timings (set-up+solution time) for $p-MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 3

| | | $p-MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^3$ | 369 | $0.73 + 2.19$ | 120 | $2.52 + 1.25$ |
| | $\alpha = 10^2$ | 267 | $0.77 + 1.60$ | 39 | $2.47 + 0.42$ |
| | $\alpha = 1$ | 45 | $0.66 + 0.26$ | 14 | $1.20 + 0.14$ |
| $\frac{1}{128}$ | $\alpha = 10^3$ | 495 | $3.52 + 11.27$ | 136 | $13.61 + 7.66$ |
| | $\alpha = 10^2$ | 300 | $3.60 + 6.93$ | 51 | $13.62 + 2.87$ |
| | $\alpha = 1$ | 42 | $3.11 + 0.80$ | 15 | $6.60 + 0.46$ |
| $\frac{1}{256}$ | $\alpha = 10^3$ | 601 | $27.29 + 75.95$ | 207 | $130.16 + 58.57$ |
| | $\alpha = 10^2$ | 323 | $27.35 + 39.92$ | 96 | $131.03 + 27.39$ |
| | $\alpha = 1$ | 42 | $25.46 + 4.79$ | 16 | $97.42 + 2.46$ |

### 3.3.4 Problem 4: discontinuous, anisotropic and full-tensor $\mathcal{C}$

Table 3.39 reports error estimates for $\alpha = 1$ for test problem 4 on rectangular meshes. The problem discontinuity causes the loss of one order of magnitude in the rate of convergence of both the potential and velocity solutions.

Interestingly, whilst the magnitude of the errors for the potential solution are comparable to those recorded for triangular meshes, the velocity errors tend to be one order of magnitude lower.

Error estimates for $\alpha = 10$ and $\alpha = 100$, are listed in Tables 3.40 and 3.41. Although first order convergence rates are also recorded, the discrete errors tend to be larger for increasing $\alpha$.

The solvers' performance for test problem 4 on rectangular meshes are reported in Table 3.42. The results of the experiments for the AMG version of the solvers is given in Table 3.43. The main findings of these two tables can be summarised as follows:

Table 3.39: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 4, $\alpha = 1$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $9.28E-03$ | – | $3.19E-02$ | – | $2.81E-02$ | – |
| $\frac{1}{32}$ | $4.88E-03$ | 0.93 | $1.67E-02$ | 0.94 | $1.43E-02$ | 0.97 |
| $\frac{1}{64}$ | $2.50E-03$ | 0.96 | $8.55E-03$ | 0.96 | $7.26E-03$ | 0.98 |
| $\frac{1}{128}$ | $1.27E-03$ | 0.98 | $4.33E-03$ | 0.98 | $3.66E-03$ | 0.99 |
| $\frac{1}{256}$ | $6.37E-04$ | 0.99 | $2.18E-03$ | 0.99 | $1.83E-03$ | 0.99 |

Table 3.40: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 4, $\alpha = 10^1$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $1.67E-02$ | – | $4.06E-01$ | – | $3.53E-01$ | – |
| $\frac{1}{32}$ | $7.76E-03$ | 1.10 | $2.14E-01$ | 0.93 | $1.81E-01$ | 0.96 |
| $\frac{1}{64}$ | $3.81E-03$ | 1.03 | $1.10E-01$ | 0.96 | $9.20E-02$ | 0.98 |
| $\frac{1}{128}$ | $1.90E-03$ | 1.01 | $5.57E-02$ | 0.98 | $4.64E-02$ | 0.99 |
| $\frac{1}{256}$ | $9.49E-04$ | 1.00 | $2.81E-02$ | 0.99 | $2.33E-02$ | 0.99 |

Table 3.41: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 4, $\alpha = 10^2$

| $h$ | $\|u - u^h\|_{L^2}$ | Rate | $\|q_x - q_x^h\|_{L^2}$ | Rate | $\|q_y - q_y^h\|_{L^2}$ | Rate |
|---|---|---|---|---|---|---|
| $\frac{1}{16}$ | $8.18E-02$ | – | $4.25E+00$ | – | $3.67E+00$ | – |
| $\frac{1}{32}$ | $2.23E-02$ | 1.88 | $2.24E+00$ | 0.92 | $1.89E+00$ | 0.96 |
| $\frac{1}{64}$ | $6.82E-03$ | 1.71 | $1.15E+00$ | 0.96 | $9.61E-01$ | 0.98 |
| $\frac{1}{128}$ | $2.52E-03$ | 1.44 | $5.84E-01$ | 0.98 | $4.84E-01$ | 0.99 |
| $\frac{1}{256}$ | $1.09E-03$ | 1.20 | $2.94E-01$ | 0.99 | $2.43E-01$ | 0.99 |

1. As for triangular meshes, the MINRES iteration count is larger for discontinuous problems than for continuous problems (see, for example, test problem 1). The same aspect is observed for $PCG_{AMG}$ but not for $PCG$;

2. The degree of the discontinuity does not affect the performance of the solvers;

3. Hence the exact version of MINRES is the most efficient solver for this type of problems. However, it should be noted that the approximated version of MINRES is also very efficient given that for rectangular meshes the AMG set-up time is relatively small;

For very fine meshes (problems with $d.o.f$ of the order of $10^6$-$10^7$) the CPU cost of exactly inverting the Schur complement becomes prohibitively expensive. Hence, approximately inverting the Schur complement using AMG should become more efficient in this context. Obviously, this consideration applies to all test problems and not only to the discontinuous case.

Table 3.42: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 4

| | | $p - MINRES$ | | $PCG$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 64 | 0.83 | 81 | $0.60 + 0.60$ |
| | $\alpha = 10^1$ | 66 | 0.94 | 80 | $0.58 + 0.57$ |
| | $\alpha = 1$ | 66 | 0.92 | 81 | $0.56 + 0.59$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 64 | 4.64 | 162 | $8.60 + 5.26$ |
| | $\alpha = 10^1$ | 66 | 4.72 | 162 | $8.61 + 5.24$ |
| | $\alpha = 1$ | 66 | 4.74 | 163 | $8.61 + 5.32$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 63 | 23.41 | 328 | $137.32 + 51.22$ |
| | $\alpha = 10^1$ | 64 | 23.48 | 326 | $135.34 + 50.45$ |
| | $\alpha = 1$ | 66 | 24.71 | 325 | $134.91 + 50.29$ |

Table 3.43: Iteration count and timings (set-up+solution time) for $p - MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 4

| | | $p - MINRES_{AMG}$ | | $PCG_{AMG}$ | |
|---|---|---|---|---|---|
| $h$ | $\alpha$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | $\alpha = 10^2$ | 70 | $0.71 + 0.39$ | 15 | $1.73 + 0.17$ |
| | $\alpha = 10^1$ | 71 | $0.71 + 0.41$ | 15 | $1.73 + 0.17$ |
| | $\alpha = 1$ | 71 | $0.69 + 0.42$ | 15 | $1.73 + 0.17$ |
| $\frac{1}{128}$ | $\alpha = 10^2$ | 70 | $3.31 + 1.41$ | 16 | $9.48 + 0.66$ |
| | $\alpha = 10^1$ | 72 | $3.34 + 1.50$ | 16 | $9.37 + 0.64$ |
| | $\alpha = 1$ | 71 | $3.18 + 1.47$ | 15 | $9.23 + 0.60$ |
| $\frac{1}{256}$ | $\alpha = 10^2$ | 70 | $25.50 + 8.11$ | 17 | $115.07 + 3.28$ |
| | $\alpha = 10^1$ | 73 | $26.26 + 8.98$ | 16 | $115.35 + 3.10$ |
| | $\alpha = 1$ | 73 | $26.22 + 8.54$ | 16 | $115.63 + 3.13$ |

### 3.3.5 Problem 5: distorted rectangular mesh

Distortion of rectangular meshes is obtained in a similar fashion to that explained for triangular meshes (see §3.2.5 and Figure 3.5). Although the mesh is distorted the finite element connectivity is regular, i.e. any node has the same number of neighboring nodes. Experiments on irregular connectivity are not reported in this thesis. Discrete error estimates for test problem 5 on structured and unstructured rectangular meshes are listed in Table 3.44.

On structured rectangular meshes the potential and velocity solutions converge with rate $\mathcal{O}(h^2)$. This confirms the results of the previous experiments (excluding discontinuous problems where velocities converge with rate $\mathcal{O}(h)$).

On distorted rectangular meshes the potential solution retains second order convergence. In contrast, the experiments show the loss of one order in the convergence rates of the velocity solutions. Thus the $x$-component of the velocity converges with rate $\mathcal{O}(h^{1.16})$ and the $y$-component with rate $\mathcal{O}(h^{1.31})$.

The loss of accuracy in velocity solutions obtained by MFEM and MHFEM on quadrilateral meshes is well-known and solutions to this issue have been proposed by Shen (1994), Arnold et al. (2005) and more recently by Younes et al. (2010), for example.

The problem lies in the fact that the Piola transformation of vectorial basis functions defined on a square reference element to the actual element is not affine for quadrilateral elements (Arnold et al. 2005). This causes loss of convergence for the approximation of the fluxes. The same situation does not occur on triangular elements.

The loss of convergence reported in Table 3.44 refers to a simple problem with unit conductivity coefficient and trivial geometry. Therefore it is expected that this would be more severe on problems with general coefficients and complex geometry.

The solvers' performance for test problem 5 on structured and unstructured meshes are reported in Tables 3.45 and 3.46. The findings of those tables are summarised as follows:

Table 3.44: $L^2(D^h)$ error estimates for the $u$, $q_x$ and $q_y$ for test problem 5

| | | | *Structured Meshes* | | | |
|---|---|---|---|---|---|---|
| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate |
| $\frac{1}{16}$ | $8.07E-05$ | – | $2.59E-04$ | – | $2.59E-04$ | – |
| $\frac{1}{32}$ | $2.02E-05$ | 2.00 | $6.47E-05$ | 2.00 | $6.47E-05$ | 2.00 |
| $\frac{1}{64}$ | $5.04E-06$ | 2.00 | $1.62E-05$ | 2.00 | $1.62E-05$ | 2.00 |
| $\frac{1}{128}$ | $1.26E-06$ | 2.00 | $4.05E-06$ | 2.00 | $4.05E-06$ | 2.00 |
| $\frac{1}{256}$ | $3.15E-07$ | 2.00 | $1.01E-06$ | 2.00 | $1.01E-06$ | 2.00 |
| | | | *Unstructured Meshes* | | | |
| $h$ | $\|u-u^h\|_{L^2}$ | Rate | $\|q_x-q_x^h\|_{L^2}$ | Rate | $\|q_y-q_y^h\|_{L^2}$ | Rate |
| $\frac{1}{16}$ | $8.54E-05$ | – | $1.93E-03$ | – | $2.03E-03$ | – |
| $\frac{1}{32}$ | $1.98E-05$ | 2.11 | $9.70E-04$ | 0.99 | $9.56E-04$ | 1.09 |
| $\frac{1}{64}$ | $5.33E-06$ | 1.89 | $4.09E-04$ | 1.25 | $4.06E-04$ | 1.23 |
| $\frac{1}{128}$ | $1.30E-06$ | 2.04 | $1.81E-04$ | 1.18 | $1.77E-04$ | 1.20 |
| $\frac{1}{256}$ | $3.33E-07$ | 1.96 | $7.85E-05$ | 1.21 | $7.85E-05$ | 1.17 |

1. The MINRES iteration count for problems with unstructured meshes is approximately twice as that for problems with structured meshes when the Schur complement is inverted exactly. For the AMG case, instead, the difference in iteration count is less marked;

2. The *PCG* iteration count also varies only slightly between structured and unstructured meshes. The same can be stated for CG with the AMG preconditioner;

3. Once again, MINRES with the exact version of preconditioner (2.41) is the best performing method.

## 3.4 Conclusions

The aim of this chapter was to report results on numerical experiments based on mixed finite element methods and compare the approximations with exact solutions. This, in addition to investigating MFEM convergence performance, allows the

Table 3.45: Iteration count and timings (set-up+solution time) for $p - MINRES$ and $PCG$ - Test problem 5

| | *Structured Meshes* | | | |
|---|---|---|---|---|
| | $p - MINRES$ | | $PCG$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 26 | 0.33 | 71 | $0.58 + 0.52$ |
| $\frac{1}{128}$ | 23 | 1.77 | 133 | $8.71 + 4.29$ |
| $\frac{1}{256}$ | 20 | 7.24 | 251 | $137.05 + 39.89$ |
| | *Unstructured Meshes* | | | |
| | $p - MINRES$ | | $PCG$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 43 | 0.54 | 81 | $0.56 + 0.59$ |
| $\frac{1}{128}$ | 42 | 3.12 | 157 | $8.75 + 5.19$ |
| $\frac{1}{256}$ | 40 | 14.76 | 279 | $136.51 + 43.62$ |

Table 3.46: Iteration count and timings (set-up+solution time) for $p - MINRES_{AMG}$ and $PCG_{AMG}$ - Test problem 5

| | *Structured Meshes* | | | |
|---|---|---|---|---|
| | $p - MINRES_{AMG}$ | | $PCG_{AMG}$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 36 | $0.68 + 0.27$ | 14 | $1.26 + 0.14$ |
| $\frac{1}{128}$ | 36 | $3.14 + 0.75$ | 15 | $6.93 + 0.55$ |
| $\frac{1}{256}$ | 36 | $25.05 + 4.00$ | 15 | $96.13 + 2.56$ |
| | *Unstructured Meshes* | | | |
| | $p - MINRES_{AMG}$ | | $PCG_{AMG}$ | |
| $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| $\frac{1}{64}$ | 48 | $0.70 + 0.35$ | 15 | $2.41 + 0.17$ |
| $\frac{1}{128}$ | 48 | $3.16 + 1.15$ | 16 | $11.74 + 0.75$ |
| $\frac{1}{256}$ | 48 | $25.18 + 5.64$ | 18 | $118.08 + 3.41$ |

validation of the codes developed in this work.

We have seen that MFEM possesses a second order convergence for the potential and first order convergence for the velocities on structured and unstructured triangular meshes. For discontinuous problems there is a loss of one order of convergence for the potential solution while the rate of convergence for the velocity solutions is unaltered. The MFEM possesses second order convergence for the potential and velocity solutions on structured rectangular meshes. The loss of approximately one order of convergence (or more is expected on complex problems) is recorded for unstructured meshes. For discontinuous problems there is a loss of one order of convergence for both the potential and velocity approximations.

The approximation for the potential tends to have spurious negative values for problems with diagonal anisotropic and full-tensor anisotropic coefficients on triangular meshes. This also occurs for problems with full-tensor anisotropic coefficients on rectangular meshes. Furthermore we have seen that, in agreement with results presented by other researchers (Younes & Fontaine 2008*b*,*a*), spurious negative oscillations are present in all cases (isotropic, anisotropic full-tensor) in test problem 3. For the isotropic case the magnitude of the oscillations tend to zero as the mesh is progressively refined.

In addition to the error analysis the chapter reports the computational cost of solving the indefinite linear system obtained with MFEM and the symmetric positive definite system obtained with MHFEM. Throughout the chapter the focus was on the robustness of the solvers with respect to the conductivity coefficient $\mathcal{C}$ and the discretisation parameter $h$.

For problems with isotropic, heterogeneous coefficients, MINRES using the exact version of the Schur complement preconditioner is the most efficient method in terms of CPU cost. This is also valid for problems with anisotropic diagonal tensors but only on rectangular meshes. In these cases, MINRES is $h$-optimal and $\mathcal{C}$-optimal. Thus solving the indefinite system is the cheapest approach to solving the mixed formulation in these special instances.

MINRES using the AMG version of the Schur complement preconditioner is also very efficient. Specifically, the number of iterations and thus the CPU cost of the solvers is significantly lower when AMG is used. However, the cost of creating the coarse grids for the AMG approximation is not negligible either for the Schur complement system or for the Lagrange multiplier system. The last is larger for the Lagrange multiplier than the Schur complement and larger on triangular meshes than on rectangular meshes.

The performance of the AMG preconditioners is also linked with the $M$-matrix condition. The Schur complement is always an $M$-matrix, hence MINRES using the AMG version of the Schur complement preconditioner will never fail. In contrast, the Lagrange multiplier system is an $M$-matrix only for problems with scalar and diagonal coefficients and triangular elements. For general coefficients and triangular elements the $M$-matrix condition does not hold. Furthermore for rectangular meshes the $M$-matrix condition does not hold in any circumstance for the SPD system. Hence, using the AMG approximation of the coefficient matrix as preconditioner for CG on rectangular meshes does not guarantee success and could potentially fail.

For general full-tensor coefficients the results are more difficult to summarise. It appears that AMG preconditioners are generally better performing. On rectangular meshes the Schur complement preconditioner (AMG version) is the cheapest approach among all those considered. The same applies to triangular meshes on fine discretisations while the AMG approximation of the Lagrange multiplier system is the cheapest on medium to coarse meshes. However, for the latter case the success of black-box AMG depends on the extent by which the $M$-matrix condition is violated. Thus, its performance is problem dependent.

The results for the test cases presented show that that generally solving the indefinite system is cheaper than solving the Lagrange multiplier system. The existing misconception that tackling the solution of an indefinite system is a source of trouble should be reconsidered. This work shows that the choice of the preconditioner for the iterative scheme is crucial in determining the success of a solver independent of whether the system is indefinite or positive-definite.

## 3.5 Relevance for the stochastic modelling

The experiments reported in this chapter and the findings associated with those are important not only in the context of deterministic modelling of groundwater flow in porous media but also to the stochastic work undertaken in the following chapters.

Stochastic modelling of groundwater flow has been traditionally associated with *Monte Carlo* methods (MCM). This approach is straightforward and it involves the implementation of a large number of sequential deterministic simulations from which statistics of the numerical solutions can be derived. It is clear that the conclusions drawn in this chapter have immediate relevance to Monte Carlo methods since their computational performance is directly proportional to the CPU cost of solving the individual deterministic system.

Monte Carlo methods are dealt with only very briefly in this thesis (see Chapter 5) while most of the attention is dedicated to other stochastic techniques which belong to the wide family of *Stochastic Galerkin* (SG) methods (see Chapters 4, 6 and 7).

Stochastic Galerkin methods require the solution of only one system of equations the size of which is considerably larger than deterministic Galerkin methods. As will be explained in the following chapter the stochastic global system of equations possesses a characteristic block structure. Generally, its solution requires preconditioners which efficiently exploit that structure. An example is the so called *mean-based* preconditioner which uses the the block diagonal components of the global system of equations.

It turns out that any fast deterministic solver can be used to invert the blocks of the leading diagonal. Hence the considerations and conclusions of this chapter will be used to select and investigate efficient solvers for SG systems.

A consideration which is worthwhile anticipating is related with set-up time required for some of the solvers used in this chapter. It has been repeatedly pointed out that the cost of factorising or constructing the AMG grids can significantly penalise the solvers overall CPU cost. For the SG systems, the block diagonal components are given by the mean stiffness matrix weighted by some polynomial basis. Thus, the set-up time for the preconditioner only involves the factorisation or AMG approximation of the mean stiffness matrix. Crucially this is performed only once.

Therefore set-up times become computationally less important in the context of SG methods when compared to the overall solution time. For MCM, instead, the factorisation or AMG approximation has to be computed for every individual simulation, thus contributing significantly to the overall CPU cost.

# Chapter 4

# Spectral Stochastic Finite Element Theory

## 4.1 Introduction

The first part of this thesis has dealt with partial differential equations (PDE) in which the input parameters (such as hydraulic conductivity) are considered to be known with certainty everywhere in the discretised domain. This approach, in which model parameters can be regarded as averaged quantities, is very easy to implement and hence widely used in applications.

In the last decade there has been a growing awareness that data used by numerical models are often dominated by uncertainty. In fact, observed data are generally scarce and this leads to modelling assumptions and data interpretation which are intrinsically uncertain. In the case of groundwater modelling, material parameters such as hydraulic conductivity are estimated locally by means of pumping / slug tests or in laboratories by means of permeameters. Although these measurements are representative only to the specific scale at which the test / experiment were undertaken, often these are extrapolated to larger scales (often of the domain's size). Even though this is more a necessity than bad practice, driven by the lack of knowledge and scarcity of data, any extrapolation of this kind is dominated by uncertainty. Furthermore it should also be remembered that the measurements themselves could be affected by errors which should be taken into account in the development of the model.

Following these considerations, the idea of quantifying the uncertainty of model parameters and passing such information to the solution of the PDE has created a vast interest in the scientific community. In such a framework the model input parameters are described as random variables and the PDE is converted to a stochastic partial differential equation (SPDE). When the SPDE is equipped by suitable boundary conditions, which can also be defined as stochastic processes, then its solution is also a stochastic process. This method allows one to present model outputs as statistical quantities, these generally being the first (mean) and second (variance) moments of the solution.

The most widely used approach to the solution of SPDE is the Monte Carlo Method (MCM). This approach is based on constructing an ensemble of realizations for the model random input parameters. The PDE is therefore solved for each realization of the ensemble and statistical quantities are obtained from the set of solutions. This approach is easy and non-intrusive, i.e the method used to discretise the PDE is not modified. Generally, MCM requires a large number of realizations to create meaningful statistics and therefore can be computationally very expensive. This limitation has lead the research community to investigate alternative methods to MCM and / or to find ways to improve the performance of MCM.

Among the alternative methods is the pioneering work of Ghanem & Spanos (2003) on the classic stochastic finite element method (SFEM) where the conductivity coefficient is described as a Gaussian process. The method was subsequently generalised to other probability distributions in the work of Xiu & Karniadakis (2002*b*) and analysed by Sudret & Der Kiureghian (2000), Deb et al. (2001), Babuška & Chatzipantelidis (2002), Babuška et al. (2004), Matthies & Keese (2005). The idea of the SFEM is to restate the SPDE as a variational problem in a similar fashion to traditional FEM formulations. However in this case, in addition to the space of deterministic functions, the space of random variables is also defined and the solution is sought in their tensor product space. One crucial aspect of this method is that the deterministic and stochastic spaces are discretised separately. Therefore the conventional finite element theory and implementations still applies and in general any Galerkin method can be used for the discretisation of the deterministic part, so that the SFEM can be generalised to the stochastic Galerkin method (SG).

Similarly to classical FEM, the discretisation of SPDE by SFEM technologies results in a single linear system of equations. However, the system is significantly larger and possesses a characteristic block structure. This aspect of the method represents a fundamental limitation. In fact, the dimension of the problem grows factorially with the number of random variables used to describe the input spatial random field. As consequence of this, the solution of high dimensional problems becomes computationally infeasible, a phenomenon known as *curse of dimensionality.* More recent technologies such as stochastic collocation (Babuška et al. 2007, Nobile et al. 2008) and multilevel Monte Carlo seem to have overcome this limitation.

The idea of multilevel Monte Carlo is to combine the concepts of multigrid technologies with traditional MCM. The acceleration in convergence is guaranteed as most of the MC simulations are carried out on the set of coarse grids and only a very limited amount of time is performed on the finer grids. Multilevel MC have been applied to the solution of ordinary differential equations (see Giles (2008), Giles & Waterhouse (2009)) and PDE (see Graham et al. (2011), Cliffe et al. (2011)). The latter papers clearly show that Multilevel MC methods are incredibly efficient for problems with rough coefficients (i.e spatial random fields with large variance or / and small correlation lengths). These types of problems, common to radioactive waste disposal applications, require a large number of random variables ($>$ 100 modes of Karhunen Loève expansion) in probability space to accurately represent the variability of the spatial random field. Their solution by SG methods is infeasible due to the curse of dimensionality, previously mentioned.

Despite the essential limitation of the method, SFEM or SG are widely used for engineering applications (a review of SFEM / SG engineering implementations is given by Stafanou (2009)). Equally we aim to show that this method can be successfully used in the context of groundwater modelling. Clearly, if for example the conductivity field is homogeneous the method can be used without any restrictions. Conversely, if the conductivity field is largely heterogeneous, such variability can be resolved by identifying areas (sub-regions) in which the material parameter has a quasi-homogeneous behaviour (which can be accurately described by a limited number of random variables). The same idea applies to spatial fields with small correlation lengths. Practically the model domain is decomposed into many sub-domains and in each sub-domain a spatial random field (using for example KLE), with different statistical parameters, is computed. In this work we follow this approach.

Assuming that the conductivity field can be accurately represented by a discontinuous random field, other challenges remain for the efficient implementation of SFEM or / and SG methods. In fact, it is crucial to use efficient solvers and preconditioners to solve the large stochastic Galerkin systems obtained from these methods.

Solution strategies depend on the choice of basis functions for the stochastic space. There are two popular choices. The first uses global complete polynomials, commonly referred to as polynomial chaos, which are orthogonal. This is the classical

SFEM approach as outlined in the original work of Ghanem & Spanos (2003). In this approach a large and highly structured linear system has to be solved. For this purpose Krylov subspace iterative solvers are a popular choice. Ghanem & Kruger (1996), Pellissetti & Ghanem (2000) proposed an efficient implementation of SFEM, without assembling the global stiffness matrix. They used a block-diagonal preconditioner (subsequently referred to as 'mean-based preconditioner') for CG based on an incomplete factorisation of the mean stiffness matrix. Powell & Elman (2009) replaced the incomplete factorisation with a black-box algebraic multigrid (AMG) solver. In Ernst et al. (2009) the implementation of the mean-based preconditioner was extended to the solution of stochastic mixed finite element systems (SMFEM). Ullmann (2008) proposed a Kronecker product preconditioner for the stochastic linear (Gaussian / uniform random fields) and non-linear (lognormal random field) cases. The implementation of the Kronecker preconditioner was recently extended to the stochastic mixed finite element method in Powell & Ullmann (2010). The preconditioner reduces significantly the number of iterations of CG and MINRES. However, its implementation is more expensive than mean-based preconditioners. A review of a large number of iterative solvers, including one-level iterative methods, multigrid methods and multilevel methods (for the stochastic discretisation) has been recently reported by Rossell & Vandewalle (2010).

The other choice uses global tensor product polynomials (Babuška et al. 2004). This implementation has the attractive advantage of allowing for the decoupling of the global Galerkin system. However, as pointed out by Ullmann (2008), this is restricted to problems in which the conductivity coefficient is approximated by normal or uniform random fields. There is no evidence that for the case in which the conductivity field is approximated by a lognormal random field (a very common assumption in the groundwater modelling community) the global Galerkin system can be decoupled.

Furthermore, it has the disadvantage that the size of the stochastic space grows more rapidly than in the complete case. Solution strategies for this choice are reviewed by Ullmann (2008) and involves iterative solvers based on Krylov subspace recycling techniques.

In this work we consider the classic SG (SFEM / SMFEM) methods based on complete orthogonal polynomials. Whilst the mixed method was extensively discussed in Chapters 2 and 3, the standard Galerkin method (FEM) was not treated. Reasons for this include the fact that its deterministic implementation has already been extensively studied and there are limitations associated with the lack of flux continuity (see Chapter 2 for further discussion). Nonetheless the stochastic implementation of standard Galerkin methods is relatively recent and it is currently a very active research area despite the aforementioned flux limitation. Therefore the discussion concerning stochastic numerical methods in groundwater flow problems reported in the following chapters focuses on both standard Galerkin and mixed finite element methods.

The methodology for the primal and mixed formulations (linear case) is reported in detail in the following sections. The derivation of the global Galerkin system is described and solution strategies that take full advantage of its characteristic block structure are proposed.

*Chapter* 5 compares numerical results for SG with those obtained by traditional MCM for a selection of test problems. This chapter is only intended to validate the SG implementation against a method which is purely based on the deterministic implementations of FEM and MFEM. The chapter does not report a thorough computational comparison of the two methods, in view of the new developments within the field of multilevel Monte Carlo methods.

Numerical results for the SG methods, linear case, are reported in *Chapter* 6 for a selection of test problems. Not only the performance of CG equipped with a mean-based preconditioner is recorded but also using the proposed block Gauss-Seidel preconditioner. The latter can also be implemented as a stand alone solver, hence its performance in those settings is also evaluated. The chapter terminates by reporting the performance of preconditioned MINRES on the same set of test problems.

The non-linear case is dealt with in *Chapter* 7. The theory partly deferring from the linear case is summarised in the initial sections. Similarly to Chapter 6, we first report the solvers performance for the primal formulation followed by results for the mixed formulation. The test problems used are similar to those described in Chapter 6. However, in this case the conductivity is approximated by a lognormal field.

The numerical implementation of the stochastic Galerkin methods has been coded by the author within the MATLAB environment and the computations are all performed in serial. The development of the same algorithms in a parallel architecture is matter for future work and development. The derivation of the polynomial chaos basis was obtained explicitly using the MATLAB symbolic toolbox in a similar manner to that explained in Ghanem & Spanos (2003). The discretization of spatial random fields using Karhunen Loève expansion method was possible adapting some MATLAB-based functions provided by Sudret & Der Kiureghian (2000).

## 4.2 The mathematical model

The steady-state flow of water in porous media, whose material parameters are assumed to be unknown, is described by a scalar second-order stochastic partial differential equation (SPDE). In the context of groundwater flow modelling the most uncertain parameter is the hydraulic conductivity. If it is not a function of the spatial variable $\mathbf{x}$, the conductivity coefficient can be represented by a set of uncorrelated random variables (*white noise* approach) $\mathcal{C}_i(\omega)$, $i = 1, \ldots, N_D$, where $N_D$ is the number of subdomains having different hydraulic properties. Alternatively, the conductivity coefficient $\mathcal{C}(\mathbf{x}, \omega)$ is a spatial random field such that for a fixed spatial location $\mathbf{x} \in D$, $\mathcal{C}(\cdot, \omega)$ is a random variable and for a fixed realization $\omega \in \Omega$, $\mathcal{C}(\mathbf{x}, \cdot)$ is a spatial field.

Let $D$ be a domain in $\mathbf{R}^d$, where $d = 2, 3$, bounded by $\Gamma = \Gamma_D \cup \Gamma_N$, as defined for the deterministic problem (see §2.2). Let $\Omega$ be the set of random events that together with the minimal $\sigma$-algebra, $\Im$, and the probability measure, $Pr$, denotes the probability space $(\Omega, \Im, Pr)$. We seek a random field solution $(u(\mathbf{x}, \omega) \in D \times \Omega)$ to the second-order elliptic problem

$$\begin{aligned} -\nabla \cdot \mathcal{C}(\mathbf{x}, \omega) \nabla u(\mathbf{x}, \omega) &= f(\mathbf{x}) && \text{in } D \times \Omega, \\ u(\mathbf{x}, \omega) &= g(\mathbf{x}) && \text{on } \Gamma_D \times \Omega, \\ \mathcal{C}(\mathbf{x}, \omega) \nabla u(\mathbf{x}, \omega) \cdot \mathbf{n} &= 0 && \text{on } \Gamma_N \times \Omega, \end{aligned} \tag{4.1}$$

where $\mathbf{n}$ denotes the unit outward normal vector to $\Gamma_N$, $g(\mathbf{x})$ represents the deterministic prescribed constant head on $\Gamma_D$ and $f(\mathbf{x})$ represents a deterministic source or sink term. Note that $f(\mathbf{x})$ and $g(\mathbf{x})$ could also be (spatial) random functions.

The solution to Problem (4.1) gives the mean potential or pressure head $u$ and the associated standard deviation, everywhere in $D$. As in the deterministic case, the potential can be used to derive the fluid discharge (flux) or Darcian velocity $\mathbf{q}$ using Darcy's Law. The limitations of this approach are highlighted in §2.2 and these apply equally to the stochastic formulation.

A more suitable approach which allows us to derive accurate approximations for the fluxes, is obtained by restating Problem (4.1) by explicitly introducing Darcy's

Law and seeking the solution $(u(\mathbf{x},\omega), \mathbf{q}(\mathbf{x},\omega)) \in D \times \Omega$ of the problem

$$\begin{aligned} \mathcal{C}^{-1}(\mathbf{x},\omega)\mathbf{q}(\mathbf{x},\omega) + \nabla u(\mathbf{x},\omega) &= 0 && \text{in } D \times \Omega, \\ \nabla \cdot \mathbf{q}(\mathbf{x},\omega) &= f(\mathbf{x}) && \text{in } D \times \Omega, \\ u(\mathbf{x},\omega) &= g(\mathbf{x}) && \text{on } \Gamma_D \times \Omega, \\ \mathbf{q}(\mathbf{x},\omega) \cdot \mathbf{n} &= 0 && \text{on } \Gamma_N \times \Omega. \end{aligned} \tag{4.2}$$

The solution to problem (4.2) gives the mean potential (or pressure) and normal fluxes and their associated standard deviations, everywhere in $D$. Problems (4.1) and (4.2) can be solved using *stochastic Galerkin* (SG) methods (Ghanem & Spanos 2003, Xiu & Karniadakis 2002*b*). In these methods the deterministic and stochastic spaces are discretised separately. Traditional Galerkin methods are used for the approximation of the deterministic space and a *polynomial chaos expansion* (PCE) is employed for the stochastic space. Hence, the theoretical definitions presented in Chapter 2 apply also to the following discussion.

## 4.3 Hydraulic Conductivity Coefficient Approximation

In this thesis we use two approaches to represent the conductivity coefficient. The first approach, herein referred to as *coloured noise*, assumes that the conductivity coefficient varies randomly from one point of $D$ to another point according to a given correlation function. Given the advantages discussed by Ghanem & Spanos (2003), Powell & Elman (2009), Deb et al. (2001), Matthies & Keese (2005) we use the Karhunen-Loève expansion (KLE) to approximate continuous and discontinuous spatial random fields.

The second approach, herein referred to as *white noise*, assumes that the conductivity coefficient varies randomly and independently from one point of $D$ to another. Although the hydraulic conductivity is spatially correlated it is common practice in applications to represent it by piecewise constant subdomains with each subdomain possessing hydraulic properties pertaining to a specific hydro-geological unit.

Since the white noise approach assumes spatial independence, thus making it theoretically unsuitable to approximate parameters such as hydraulic conductivity, it has been rarely used in the research community. In fact the literature is rich on the implementation of SG techniques using KLE methods (see Xiu & Karniadakis (2002*a*), Ghanem & Dham (1998)).

*Coloured Noise.* The conductivity field $\mathcal{C}(\mathbf{x}, \omega)$, is approximated by a truncated Karhunen-Loève expansion (Loève 1977)

$$\mathcal{C}(\mathbf{x}, \omega) \approx \mathcal{C}_d^h(\mathbf{x}, \xi(\omega)) = \mu(\mathbf{x}) + \sigma \sum_{i=1}^{d} \sqrt{\lambda_i} \xi_i \beta_i(\mathbf{x}), \tag{4.3}$$

where $\beta_i(\mathbf{x})$ and $\lambda_i$ are the eigenfunctions and eigenvalues of the covariance function and are obtained from the solution of the eigenvalue problem

$$\int_D \rho(\mathbf{x}, \mathbf{x}') \beta_i(\mathbf{x}') dD_{\mathbf{x}'} = \lambda_i \beta_i(\mathbf{x}), \qquad i = 1, \ldots, \infty, \tag{4.4}$$

$\xi_i$ are random variables, $\mu(\mathbf{x})$ and $\sigma$ are the mean and the standard deviation of $\mathcal{C}(\mathbf{x}, \omega)$, respectively. $\rho(\cdot, \cdot)$ is the correlation function of the spatial random field $\mathcal{C}$. When the conductivity coefficient $\mathcal{C}(\cdot, \omega)$ is assumed to be a Gaussian process, the random variables $\xi_i$ in (4.3) are normally distributed. In these circumstances the random variables have the desirable property of being uncorrelated and independent. However this also makes problems (4.1) and (4.2) ill-posed since the diffusion coefficient is not bounded below and above by positive constants (Powell & Elman 2009). In fact, it is well known that the conductivity field is required to be strictly positive and bounded, i.e.

$$0 < k_1 \leq \mathcal{C}(\mathbf{x}, \omega) \leq k_2 < +\infty. \tag{4.5}$$

Although Gaussian functions possess an infinite spectrum, it can be shown that well-defined discrete solutions can be obtained if a relatively small variance is used .

Alternatively, condition (4.5) can be satisfied by transforming the Gaussian random field into a lognormal one by expanding the $d$-terms of the KLE into the polynomial chaos of order less than or equal to $p$. This procedure was proposed by several authors (see Ghanem & Spanos (2003), Sudret & Der Kiureghian (2000)). However, this implementation has some important drawbacks. For example, the coefficient matrix arising from the discretisation of (4.1) and (4.2) using lognormal random fields becomes block dense and ill-conditioned, which makes the linear system very difficult to solve.

Xiu & Karniadakis (2002*a*) have used uniform random variables, hence ensuring that $\mathcal{C}_d^h(\mathbf{x}, \omega)$ is bounded between two positive values with probability 1. A consequence of this approach is that the random variables in (4.3) are not guaranteed to be independent, thus this condition needs to be assumed explicitly (Xiu & Karniadakis 2002*b*). Other distributions, such as the Gamma and Beta distributions, can be employed. In this thesis we use random variables which are assumed to be either uniformly or normally distributed. A separate discussion is provided for lognormal distributions.

Different statistical parameters can be assigned to different regions of $D$, thus reflecting the diverse hydraulic behaviour of natural deposits. We perform a coarse subdomain decomposition of $D$ and define a continuous random field for each $D_k$, $k = 1, \ldots, N_D$, such that

$$\mathcal{C}_D^h(\mathbf{x}, \xi(\omega)) = \bigcup_{k=1}^{N_D} \mathcal{C}_{D_k}^h(\mathbf{x}, \xi(\omega)),$$

where $N_D$ is the number of sub-domains in $D$. Now, each sub-domain $D_k$, which may be of irregular shape, is enclosed by a rectangular-shaped domain $D_k^{'}$ such that,

$D_k \subset D'_k$, for $k = 1, \dots, N_D$. The 'fictional' domain $D'_k$ can be the smallest rectangle enclosing $D_k$ or can be larger than $D_k$. A Karhunen-Loève expansion is implemented for each sub-domain $D_k$ but the eigenvalue problem (4.4) is solved with respect to $D'_k$. The hydraulic conductivity discontinuous random field is defined as

$$\mathcal{C}^h(\mathbf{x}, \xi(\omega)) = \bigcup_{k=1}^{N_D} \left[ \mu_k(\mathbf{x}) + \sigma_k \sum_{i=1}^{d_k} \sqrt{\lambda_k^i} \xi_k^i \beta_k^i(\mathbf{x}) \right]. \tag{4.6}$$

When the exponential correlation function and a square / rectangular domain are considered, there exists closed form solutions to the eigenvalue problem (4.4) (Ghanem & Spanos 2003). In this thesis we make full use of the closed form solutions, thus only random fields whose correlation function is of exponential or square-exponential type are considered. Examples in which the eigenvalue problem is solved numerically can be found in Lu & Zhang (2007) and description of numerical algorithms are reported in Ghanem & Spanos (2003). Note that in such cases the computational cost of solving the eigenproblem (4.4) needs to be evaluated.

*White Noise.* The white noise approach is often used to approximate parameters such as rainfall or groundwater recharge which (generally) do not show strong spatial correlation. Although the hydraulic conductivity is a function of $\mathbf{x}$, in practice a very complex spatial distribution can always be reduced to a (finite) set of subdomains with constant parameter values.

From a mathematical point of view the white noise approach has significant advantages with respect to KLE based approaches. We will see in the next sections that the linear systems obtained with this approach have a (favourable) block-tridiagonal structure (Constantine 2009). Hence, block diagonal preconditioners can be efficiently used to solve these problems.

The conductivity field can be defined as follow

$$\mathcal{C}(\cdot, \xi) = \bigcup_{i=1}^{N_D} k_i(\xi). \tag{4.7}$$

where $k_i(\xi)$, $i = 1, \ldots, N_D$, is a set of random variables. These could be normally or uniformly distributed (other distributions are also possible). For the case in which $k_i(\xi)$ are uniformly distributed, these have the form

$$k(\xi) = \frac{\xi\,(k_2 - k_1) + (k_2 + k_1)}{2}, \qquad \xi \sim U[-1, 1] \tag{4.8}$$

where $k_1$ and $k_2$ are defined as in (4.5) and $\xi$ are uniform random variables defined in the interval $[-1, 1]$.

Depending on the choice of random variables the basis functions of the probability space are chosen so that they are orthogonal with respect to the probability measure associated with the random variables. For example, in the case of uniform random variables, the basis functions are univariate Legendre polynomials (for the case in which the KLE is implemented, the basis functions are multi-dimensional Legendre polynomials). If normal random variables are used the basis comprises univariate or multivariate Hermite polynomials depending on the approach that is used to approximate the conductivity field. A list of Wiener-Chaos polynomial bases and the underlying random variables (including their support) is given in Xiu & Karniadakis (2002*b*,*a*).

## 4.4 The weak formulation

The weak formulation of problem (4.1) is given by Powell & Elman (2009), while that for the mixed formulation (4.2) is given by Furnival (2008). We briefly summarise this derivation in the following sub-sections. Although the treatment is somewhat technical, this is needed for a complete presentation of the topic.

Before stating the weak formulation of problems (4.1) and (4.2), some considerations regarding random variables are required. Suppose that $X$ is a random variable defined in $(\Omega, \Im, Pr)$ and denoting the density function by $f_X(x)$, we can express the mathematical expectation as

$$\langle X \rangle = \int_\Omega X dP = \int_{\mathbb{R}} x f_X(x) dx. \tag{4.9}$$

Similarly, for a finite set of random variables $\{\xi_1, \ldots, \xi_d\} \in \Omega$, we can define a function $g(y)$, so that

$$\langle g(\boldsymbol{\xi}) \rangle = \int_\Omega g(\boldsymbol{\xi}) dP = \int_{\Xi^d} g(y) f_{g(\boldsymbol{\xi})}(y) dy, \tag{4.10}$$

where $f_{g(\boldsymbol{\xi})}(y)$ is the joint probability density function of the random variables, $\Xi \subset \mathbb{R}$, and $y \in \mathbb{R}^d$.

### 4.4.1 Primal Formulation

The weak formulation of the primal variational problem is: find $u \in W$ such that

$$\langle a(u, w) \rangle = \langle L(w) \rangle, \qquad \forall w \in W \tag{4.11}$$

where

$$\begin{aligned} \langle a(u,w) \rangle &= \int_\Omega \left[ \int_D K(\mathbf{x}, \boldsymbol{\xi}) \nabla u(\mathbf{x}, \boldsymbol{\xi}) \cdot \nabla w(\mathbf{x}, \boldsymbol{\xi}) dD \right] dP, \\ \langle L(w) \rangle &= \int_\Omega \left[ \int_D f(\mathbf{x}) w(\mathbf{x}, \boldsymbol{\xi}) dD \right] dP. \end{aligned} \tag{4.12}$$

The solution space $W$ is the tensor product space

$$W = H_0^1(D) \otimes L^2(\Omega), \tag{4.13}$$

where the subspace $H_0^1(D)$ is defined

$$H_0^1(D) = \{w : w \in H^1(D) \text{ and } w = 0 \text{ on } \Gamma\}, \tag{4.14}$$

and

$$\begin{aligned} H^1(D) &= \{w : w \in L^2(D) \text{ and } \tfrac{\partial w}{\partial x_i} \in L^2(D), i = 1, \ldots, d\}, \\ L^2(D) &= \{w : \ w \text{ is defined on } D \text{ and } \int_D w^2 dD < \infty\}, \\ L^2(\Omega) &= \{w : \ w \text{ is defined on } \Omega \text{ and } \int_\Omega w^2 d\Omega < \infty\}. \end{aligned} \tag{4.15}$$

The *Lax-Milgram* lemma can be used to prove that there exists a unique solution to this problem provided that condition (4.5) is satisfied.

### 4.4.2 Mixed Formulation

The weak formulation of the mixed variational problem is: find $(u, \mathbf{q}) \in V \times W$

$$\begin{aligned} \langle a(\mathbf{q},\mathbf{v})\rangle + \langle b(\mathbf{v},u)\rangle &= \langle (g, \mathbf{n}\cdot\mathbf{v})_{\Gamma_D}\rangle, && \forall \mathbf{v} \in V \\ \langle b(\mathbf{q},w)\rangle &= -\langle (f,w)\rangle, && \forall w \in W \end{aligned} \tag{4.16}$$

where

$$\begin{aligned} \langle a(\mathbf{q},\mathbf{v})\rangle &= \int_\Omega \left[\int_D \frac{1}{K(\mathbf{x},\boldsymbol{\xi})}\mathbf{q}(\mathbf{x},\boldsymbol{\xi})\cdot\mathbf{v}(\mathbf{x},\boldsymbol{\xi})dD\right]dP, \\ \langle b(\mathbf{v},w)\rangle &= \int_\Omega \left[\int_D \nabla\cdot\mathbf{v}(\mathbf{x},\boldsymbol{\xi})w(\mathbf{x},\boldsymbol{\xi})dD\right]dP, \\ \langle (g,\mathbf{n}\cdot\mathbf{v})_{\Gamma_D}\rangle &= \int_\Omega \left[\int_{\Gamma_D} g(\mathbf{x})(\mathbf{n}\cdot\mathbf{v}(\mathbf{x},\boldsymbol{\xi}))dD\right]dP, \\ \langle (f,w)\rangle &= \int_\Omega \left[\int_D f(\mathbf{x})w(\mathbf{x},\boldsymbol{\xi})dD\right]dP. \end{aligned} \tag{4.17}$$

The solution spaces $W = L^2(D) \otimes L^2(\Omega)$, where $L^2(D)$ and $L^2(\Omega)$ are defined in (4.15). The solution space $V$ is the tensor product space

$$V = \{\mathbf{v}(\mathbf{x},\boldsymbol{\xi}) \in H(div; D) \otimes L^2(\Omega) : \mathbf{v}(\mathbf{x},\boldsymbol{\xi})\cdot\mathbf{n} = 0 \text{ on } \Gamma_N \times \Omega\}, \tag{4.18}$$

where, given the vector function $\mathbf{v} = \{v_1, \ldots, v_d\}$,

$$H(div; D) = \{\mathbf{v} : \mathbf{v} \in L^2(D)^d, \text{ and } \nabla\cdot\mathbf{v} \in L^2(D)\}, \tag{4.19}$$

and $L^2(D)^d$ is the *Hilbert* space

$$L^2(D)^d = \{\mathbf{v} : v_i \in L^2(D), i = 1, \ldots, d\}. \tag{4.20}$$

There exists a unique solution to this problem providing that the bilinear forms are continuous and coercive and the *inf-sup* inequality is satisfied (see Brezzi & Fortin

(1991)). Furthermore condition (4.5) needs to be satisfied, i.e.

$$0 < \frac{1}{k_1} \leq \frac{1}{K(\mathbf{x},\omega)} \leq \frac{1}{k_2} < +\infty. \tag{4.21}$$

## 4.5 Stochastic Finite Element Approximation

The implementation of the spectral stochastic finite element method (SSFEM) for problem (4.1) involves the separate discretisation of the deterministic and stochastic spaces. The deterministic space $H_0^1(D)$ is discretised by means of polynomials defining the classical finite element basis functions $\phi_i(\mathbf{x})$, $i = 1, \ldots, N_u$, where $N_u$ is the number of finite element nodes. These basis functions are piecewise linear on a partition $Z^h$ of $D$ defined by triangular finite elements $\triangle_i$, $i = 1, \ldots, N_e$, such that,

$$Z^h = \bigcup_{i=1}^{N_e} \triangle_i,$$

where $N_e$ denotes the number of finite elements. Here $h$ denotes the discretisation parameter and describes the size of the finite elements in $Z^h$. Let $E^h$ be the collection of numbered edges $(\mathcal{D} = 2)$, $e_i$, $i = 1, \ldots, N_{edg}$, where $N_{edg}$ is the total number of edges in $Z^h$.

The stochastic space $L^2(\Omega)$ is discretised by means of polynomial chaos of order less than or equal to $p$ in $d$ random variables $\xi_i$. According to the Galerkin method we define the finite dimensional subspaces $S^h \subset H_0^1(D)$ and $T^h \subset L^2(\Omega)$ such that $W^h = S^h \otimes T^h \subset W = H_0^1(D) \otimes L^2(\Omega)$. The discrete variational formulation of (4.11) is: Find $u^h \in W^h$ such that

$$\langle a(u^h, w^h) \rangle = \langle L(w^h) \rangle. \qquad \forall w^h \in W^h \tag{4.22}$$

### 4.5.1 Polynomial Chaos

The basis for subspace $T^h$ contains multidimensional polynomials of degree less than or equal to $p$, $T^h = span\{\chi_i, \ldots, \chi_P\}$ where

$$P = \frac{(d+p)!}{d!p!}, \tag{4.23}$$

and $d$ represents the number of random variables (number of terms retained in the KLE expansion). The *polynomial chaos* basis is chosen so that the following orthogonality condition is satisfied

$$\langle \chi_i \chi_j \rangle = \langle \chi_i \rangle^2 \delta_{i,j}. \tag{4.24}$$

In this thesis the probability measure corresponds to that of either a $d$-dimensional uniform distribution or $d$-dimensional normal distribution. Hence, the basis for $T^h$ consists of $d$-dimensional Legendre or Hermite polynomials. Note that the one-dimensional case is a special form of these larger spaces.

#### Legendre Polynomials

Multidimensional Legendre polynomials are defined as products of univariate Legendre polynomials, $\{L_i(\xi_j)\}$, $i = 0, \ldots, p$ and $j = 1, \ldots, d$. Let us associate to each basis function $\{\chi_i\}$, $i = 1, \ldots, P$, a multi-index $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{(i,j)}$, where the components represent the degree of the univariate polynomials $\{L_i(\xi_j)\}$. For example, given the univariate Legendre polynomials of degree less than or equal to 3, we have

$$L_0(\xi) = 1 \qquad L_1(\xi) = \xi \qquad L_2(\xi) = \frac{1}{2}\left(3\xi^2 - 1\right) \qquad L_3(\xi) = \frac{1}{2}\left(5\xi^3 - 3\xi\right), \tag{4.25}$$

and considering two-dimensional polynomials, i.e. $d = 2$, we have the indices $\alpha_k$, $k = 1, \ldots, 10$

$$\begin{aligned} \alpha_1 &= \alpha_{(0,0)} \qquad \alpha_2 = \alpha_{(1,0)} \qquad \alpha_3 = \alpha_{(0,1)} \qquad \alpha_4 = \alpha_{(2,0)} \qquad \alpha_5 = \alpha_{(1,1)} \\ \alpha_6 &= \alpha_{(0,2)} \qquad \alpha_7 = \alpha_{(3,0)} \qquad \alpha_8 = \alpha_{(2,1)} \qquad \alpha_9 = \alpha_{(1,2)} \qquad \alpha_{10} = \alpha_{(0,3)}. \end{aligned} \tag{4.26}$$

The basis functions defined in subspace $T^h = \{\chi_{\alpha_1}, \dots, \chi_{\alpha_{10}}\}$ are

$$\begin{aligned}
\chi_{\alpha_1} &= 1 & \chi_{\alpha_2} &= \xi_1 & \chi_{\alpha_3} &= \xi_2 \\
\chi_{\alpha_4} &= \tfrac{1}{2}\left(3\xi_1^2 - 1\right) & \chi_{\alpha_5} &= \xi_1\xi_2 & \chi_{\alpha_6} &= \tfrac{1}{2}\left(3\xi_2^2 - 1\right) \\
\chi_{\alpha_7} &= \tfrac{1}{2}\left(5\xi_1^3 - 3\xi_1\right) & \chi_{\alpha_8} &= \tfrac{1}{2}(3\xi_1^2 - 1)\xi_2 & \chi_{\alpha_9} &= \tfrac{1}{2}\xi_1(3\xi_2^2 - 1) \\
\chi_{\alpha_{10}} &= \tfrac{1}{2}\left(5\xi_2^3 - 3\xi_2\right).
\end{aligned} \tag{4.27}$$

### Hermite Polynomials

Similarly, multidimensional Hermite polynomials are obtained as a product of univariate Hermite polynomials, $\{H_i(\xi_j)\}$, $i = 0, \dots, p$ and $j = 1, \dots, d$. Following the previous example, the univariate Hermite polynomials of degree less than or equal to 3 are given by

$$H_0(\xi) = 1 \qquad H_1(\xi) = \xi \qquad H_2(\xi) = \xi^2 - 1 \qquad H_3(\xi) = \xi^3 - 3\xi. \tag{4.28}$$

Considering the indices (4.26), the basis functions for the stochastic space $T^h$ are given by

$$\begin{aligned}
\chi_{\alpha_1} &= 1 & \chi_{\alpha_2} &= \xi_1 & \chi_{\alpha_3} &= \xi_2 \\
\chi_{\alpha_4} &= \xi_1^2 - 1 & \chi_{\alpha_5} &= \xi_1\xi_2 & \chi_{\alpha_6} &= \xi_2^2 - 1 \\
\chi_{\alpha_7} &= \xi_1^3 - 3\xi_1 & \chi_{\alpha_8} &= \xi_2(\xi_1^2 - 1) & \chi_{\alpha_9} &= \xi_1(\xi_2^2 - 1) \\
\chi_{\alpha_{10}} &= \xi_2^3 - 3\xi_2.
\end{aligned} \tag{4.29}$$

## 4.5.2 Linear System

To obtain the discrete linear system associated with the weak formulation (4.22) the potential $u^h$ is approximated by

$$u^h(\mathbf{x}, \xi) = \sum_{s=1}^{P} \sum_{r=1}^{N_u} u_{r,s}\phi_r(\mathbf{x})\chi_s(\xi). \tag{4.30}$$

Substituting $u^h$ using expansion (4.30) in (4.22) we obtain the linear system of equations

$$A\mathbf{u} = \mathbf{f}, \tag{4.31}$$

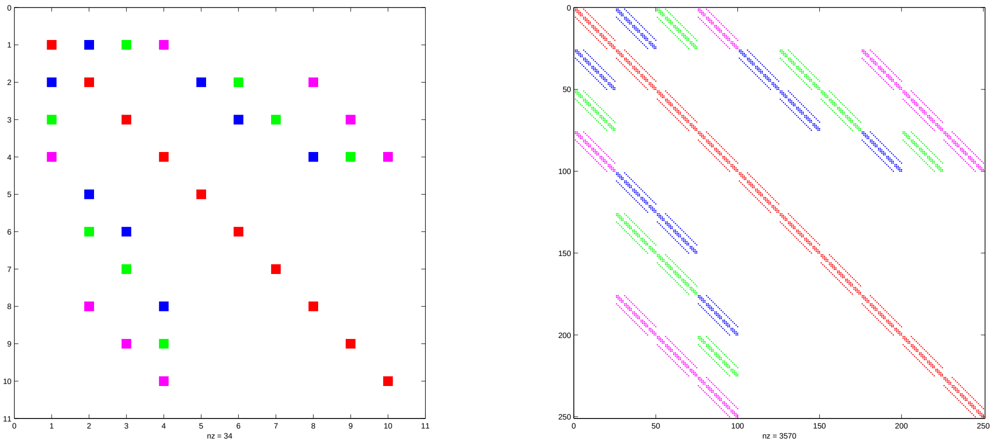
where $A$ is a sparse matrix of size $N_u P \times N_u P$ with a block-structure

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,P} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ A_{P,1} & A_{P,2} & \cdots & A_{P,P} \end{bmatrix}, \tag{4.32}$$

and

$$\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_P]^T, \qquad \mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_P]^T. \tag{4.33}$$

The block structure of $A$ has been described by other authors (see Powell & Elman (2009), for example). We include a brief description for completeness. Consider the case $p = 2$ and $d = 3$, then $P = 10$. The structure of $A$ for this example is illustrated in Figure 4.1.



(a) Block structure of $A$ (b) Sparsity pattern of $A$

Figure 4.1: Block structure of the global stiffness matrix $A$ for the second order problem

The diagonal blocks $A^*$ (red squares) are defined as products of the mean stiffness matrix, $K_0$, and $\langle\chi_i\rangle^2$ so that the $i$th diagonal block, $A^*_{i,i}$, is

$$A^*_{i,i} = \langle\chi_i\rangle^2 \otimes K_0, \qquad i = 1, \ldots, P, \tag{4.34}$$

where

$$(K_0)_{r,s} = \int_D \mu \nabla\phi_r(\mathbf{x})\nabla\phi_s(\mathbf{x})d\mathbf{x}, \tag{4.35}$$

and $\mu$ denotes the mean value of the conductivity field $\mathcal{C}(\mathbf{x}, \xi)$.

The off-diagonal blocks $A^\star_{i,j}$, $i \neq j$, are products of the stiffness matrices, $K_l$, with the coefficients of the polynomial chaos expansion, $c_{i,j,l} = \langle\xi_l\chi_i\chi_j\rangle$, $i, j = 1, \ldots, P$ and $l = 1, \ldots, d$

$$A^\star_{i,j} = \sum_{l=1}^{d} [\langle\xi_l\chi_i\chi_j\rangle] \otimes K_l, \tag{4.36}$$

where

$$(K_l)_{r,s} = \sigma\sqrt{\lambda_l} \int_D \beta_l(\mathbf{x})\nabla\phi_r(\mathbf{x})\nabla\phi_s(\mathbf{x})d\mathbf{x}. \tag{4.37}$$

The coefficient matrix $A$ can be expressed in matrix notation. Following Powell & Elman (2009) we have

$$A = G_0 \otimes K_0 + \sum_{k=1}^{d} G_k \otimes K_k \tag{4.38}$$

Note that in Figure 4.1a different colours are assigned to different blocks of $A$. Each colour represents the tensor product operation of a stochastic matrix $G_k$ with $K_k$, $k = 0, \ldots, d$.

It is evident that the sparsity of the global stochastic coefficient matrix $A$ is governed by the coefficients of the polynomial chaos expansion. The sparsity of the blocks of $A$ is determined by the sparsity of the deterministic finite element stiffness matrix. Figure 4.1b shows the sparsity of $A$ for the case in which $h = \frac{1}{4}$.

For the case where the conductivity coefficient is approximated by (4.8), the basis for $T^h$ consists of one-dimensional Legendre polynomials of degree less than or equal to $p$. In this particular case $A$ has size $N_u p \times N_u p$ and has the following (tridiagonal) structure,

$$A = \begin{bmatrix} A_1^* & A_2^\star & & & \\ A_2^\star & A_2^* & A_3^\star & & \\ & \ddots & \ddots & \ddots & \\ & & A_{p-1}^\star & A_{p-1}^* & A_p^\star \\ & & & A_p^\star & A_p^* \end{bmatrix}. \tag{4.39}$$

The diagonal blocks $A^*$ are given by (4.34) and the off diagonal blocks $A^\star$ take the form

$$A_{i,j}^\star = \langle k \chi_i \chi_j \rangle \otimes K, \tag{4.40}$$

where

$$K(r,s) = \int_D \nabla \phi_r(\mathbf{x}) \nabla \phi_s(\mathbf{x}) d(\mathbf{x}), \tag{4.41}$$

and $k$ is as defined in (4.7).

### 4.5.3 Implementation and Solution Strategies

The global coefficient matrix $A$ is never fully assembled. In fact, its dimension grows quickly with the order of the polynomial basis $p$ and the number of random variables $d$ making its full assembling unfeasible from a memory point of view. As originally observed by Ghanem & Kruger (1996), it is necessary to store $d+1$ matrices of size $N_u \times N_u$ corresponding to $K_k$, $k = 0, ..., d$, in (4.38) and the non-zero entries of the stochastic matrices $G_k$. Depending on the size of the problem these can be stored either on RAM or disk.

The discrete linear system can be solved by the conjugate gradient method $CG$. However, most often a preconditioner, $\mathcal{P}$, is required to increase the efficiency of the solver. The earlier attempts of Ghanem & Kruger (1996) and Pellissetti & Ghanem

(2000) involved incomplete factorization schemes for the diagonal blocks of $A$. Using the notation of §4.5.2, we can define the block-diagonal preconditioner $\mathcal{P}_{bdiag}$ and the mean preconditioner $\mathcal{P}_{mean}$ as

$$\mathcal{P}_{bdiag} = G_0 \otimes K_0, \qquad \mathcal{P}_{mean} = I \otimes K_0. \tag{4.42}$$

At each CG iteration the computation of $\mathcal{P}^{-1}\mathbf{r}$ is required, where $\mathbf{r}$ is the residual vector. This operation involves the solution of $P$ sub-systems of equations (within the action of the preconditioner) of size $N_u \times N_u$ with coefficient matrix $K_0$.

As pointed out by Powell & Elman (2009) any efficient deterministic solver can be used for the solution of the $P$ sub-systems. These authors proposed the use of one V-cycle of black-box algebraic multigrid (AMG). The crucial advantage of using black-box AMG is that the computational cost of one V-cycle of AMG is linearly proportional to the discretisation parameter $h$.

It is observed (see Powell & Elman (2009)) that, when Gaussian random variables are employed, the preconditioned system is positive definite only when the variance and the order of the polynomials is small. This is a consequence of the infinite support of the Gaussian distribution and the violation of condition (4.5) for the conductivity coefficient. Preconditioned CG breaks down when this criteria is violated. Therefore, the use of Hermite polynomials is limited to problems with small variances. Uniform distributions, however, have finite support and condition (4.5) can be easily satisfied.

For the SFEM method to be computationally efficient and competitive with respect to traditional sampling methods, the CG method needs to be equipped with robust preconditioners which are optimal with respect to $h$, $d$, $p$ and especially $\mathcal{C}$. It is well known that the performance of preconditioners (4.42) (see the numerical experiments presented in Chapter 6) deteriorates for problems in which $\mathcal{C}$ has a large standard deviation. This is due to the fact that the off-diagonal blocks of $A$ become increasingly important (for large variances) and they are not included in the preconditioner, $\mathcal{P}$.

To overcome this important limitation a new preconditioner which fully exploits the block structure of $A$ is proposed. At each CG iteration the computation of $\mathcal{P}^{-1}\mathbf{r}$ involves all blocks of the coefficient matrix $A$. This is achieved by adding an internal loop to the preconditioning operation which essentially implements a full inversion of the global stiffness matrix $A$ using a block symmetric Gauss-Seidel algorithm. The preconditioner, which to the author's knowledge has not been used in the SFEM context before, takes the form,

$$\mathcal{P}_{bSGS} \approx G_0 \otimes K_0 + \sum_{k=1}^{d} G_k \otimes K_k. \tag{4.43}$$

However, the preconditioner $\mathcal{P}_{bSGS}$ is neither assembled nor inverted directly. An example should make this process more clear. Let us consider the case in which $d = 2$ and $p = 2$. The global stiffness matrix takes the form

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & 0 & A_{2,4} & A_{2,5} & 0 \\ A_{3,1} & 0 & A_{3,3} & 0 & A_{3,5} & A_{3,6} \\ 0 & A_{4,2} & 0 & A_{4,4} & 0 & 0 \\ 0 & A_{5,2} & A_{5,3} & 0 & A_{5,5} & 0 \\ 0 & 0 & A_{6,3} & 0 & 0 & A_{6,6} \end{bmatrix}.$$

At each $CG$ iteration, we iterate over $k = 1, 2, 3, \ldots$ and progressively solve the system of equations block by block for $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_6$, as follows

$$\begin{bmatrix} A_{1,1} & 0 & 0 & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & 0 & 0 & 0 & 0 \\ A_{3,1} & 0 & A_{3,3} & 0 & 0 & 0 \\ 0 & A_{4,2} & 0 & A_{4,4} & 0 & 0 \\ 0 & A_{5,2} & A_{5,3} & 0 & A_{5,5} & 0 \\ 0 & 0 & A_{6,3} & 0 & 0 & A_{6,6} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1^k \\ \mathbf{z}_2^k \\ \mathbf{z}_3^k \\ \mathbf{z}_4^k \\ \mathbf{z}_5^k \\ \mathbf{z}_6^k \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{r}_4 \\ \mathbf{r}_5 \\ \mathbf{r}_6 \end{bmatrix} - \begin{bmatrix} 0 & A_{1,2} & A_{1,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{2,4} & A_{2,5} & 0 \\ 0 & 0 & 0 & 0 & A_{3,5} & A_{3,6} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1^{k-1} \\ \mathbf{z}_2^{k-1} \\ \mathbf{z}_3^{k-1} \\ \mathbf{z}_4^{k-1} \\ \mathbf{z}_5^{k-1} \\ \mathbf{z}_6^{k-1} \end{bmatrix}, \tag{4.44}$$

where $\mathbf{z}^0$ is an initial guess and $\mathbf{r}$ is the residual vector obtained at each $CG$ iteration. The terms at the $(k-1)$th level are known from the previous iteration and hence they become part of the right hand side of the system of equations. The $k$ terms are obtained successively by solving $P$ sub-problems of size $N_u \times N_u$ using either *UMFPACK* or one V-cycle of $AMG$ or any fast solver for the deterministic problem under consideration.

The internal Gauss-Seidel iteration has to be symmetric to be used as a preconditioner for $CG$. Hence the forward sweep illustrated in (4.5.3) needs to be followed by a backward sweep. Hence for each iteration of the Gauss-Seidel algorithm, one sweep in each direction is required to guarantee the symmetry of the preconditioner for CG.

Two stopping criteria are used for the proposed algorithm. Ideally the iterative method stops when

$$\| \mathbf{z}^k - \mathbf{z}^{k-1} \|_\infty < \epsilon$$

where $\epsilon = 10^{-8}$. Alternatively, when a maximum number of iterations *maxitb* is achieved the current approximation for $\mathbf{z}$ is the preconditioned residual needed within the current CG iteration.

The algorithm used in the numerical experiments reported in Chapters 6 and 7 is given below. Algorithm 1 shows the forward sweep and 2 the backward one. When Gauss-Seidel is used as a preconditioner for $CG$ both sweeps are used and the convergence test is carried out only at the end of the backward sweep. Note that in the presented algorithm we have used the non-zero blocks of $A$ as **input** in order to simplify its description. However in the actual implementation the blocks of $A$ are computed every time. In fact, only $d+1$ ($K_0$ and $K_k$) stiffness matrices and the stochastic matrices $G_k$ are stored.

In general, this algorithm should decrease the number of CG iterations and, in particular, it should improve the iteration count for those problems for which the off-diagonal blocks of $A$ are important, i.e. problems in which the spatial random field has a large standard deviation.

**Algorithm 1** Gauss-Seidel forward sweep

```
input: $A_{i,j}$, $i = j = 1, \ldots, P$ {Non-zero blocks of $A$}
input: $\mathbf{r}_i$, $i = 1, \ldots, P$ {CG residual vector}
repeat
  input: $\mathbf{z}_j$, $j = 1, \ldots, P$ {Initial guess}
  for $i = 1$ to $P$ do
    for $j = 1$ to $i - 1$ do
      $\mathbf{rhs}_i = \mathbf{r}_i - A_{i,j}\mathbf{z}_j^k$
    end for
    for $j = i + 1$ to $P$ do
      $\mathbf{rhs}_i = \mathbf{r}_i - A_{i,j}\mathbf{z}_j^{k-1}$
    end for
    $A_{i,i}\mathbf{z}^k = \mathbf{rhs}_i$ {Solve with UMFPACK or one V-cycle of AMG code}
  end for
until convergence or $maxitb$ is reached
```

**Algorithm 2** Gauss-Seidel backward sweep

```
input: $\mathbf{z}_j^k$, $j = 1, \ldots, P$ {Vector obtained from forward sweep}
for $i = P$ to 1 do
  for $j = 1$ to $i - 1$ do
    $\mathbf{rhs}_i = \mathbf{r}_i - A_{i,j}\mathbf{z}_j^k$
  end for
  for $j = i + 1$ to $P$ do
    $\mathbf{rhs}_i = \mathbf{r}_i - A_{i,j}\mathbf{z}_j^{k+1}$
  end for
  $A_{i,i}\mathbf{z}^{k+1} = \mathbf{rhs}_i$ {Solve with UMFPACK or one V-cycle of AMG code}
end for
```

On the other hand, it is clear from the presented algorithm that the number of matrix-vector operations increases significantly. Thus, to improve the computational cost of the solution process we seek a substantial reduction in the number of $CG$ iterations with the aid of a small number of internal Gauss Seidel iterations.

The performance of $\mathcal{P}_{bSGS}$ and its comparison with mean-based preconditioners (4.42) is reported in Chapters 6 and 7.

## 4.6 Stochastic Mixed Finite Element Approximation

The approach to SMFEM is similar to the one presented in the previous section. However, the mixed finite element approximation requires the definition of subspaces for $H(div; D)$ in addition to $L^2(D)$. In this we consider the Raviart-Thomas space of lowest order $RT_0$ as a suitable space for the approximation of the velocity solution and $M_0(K)$ is defined to be the space of piecewise constant functions. These are defined in §2.5.1.

As previously presented (see §4.5.1), the stochastic space $L^2(\Omega)$ is discretised by means of polynomial chaos. The spaces for the stochastic approximation are consequently given by $V^h = Y^h \otimes T^h \subset V = H(div; D) \otimes L^2(\Omega)$ and $W^h = X^h \otimes T^h \subset W = L^2(D) \otimes L^2(\Omega)$.

The discrete variational formulation of (4.16) is: find $\mathbf{q}^h \in V^h$ and $u^h \in W^h$ such that

$$\begin{aligned} \langle a(\mathbf{q}^h, \mathbf{v}^h)\rangle + \langle b(\mathbf{v}^h, u^h)\rangle &= \langle (g, \mathbf{n}\cdot\mathbf{v})_{\Gamma_D}\rangle, && \forall \mathbf{v}^h(\mathbf{x}, \boldsymbol{\xi}) \in V^h \\ \langle b(\mathbf{q}^h, w^h)\rangle &= -\langle (f, w^h)\rangle. && \forall w^h(\mathbf{x}, \boldsymbol{\xi}) \in W^h \end{aligned} \tag{4.45}$$

### 4.6.1 Linear System

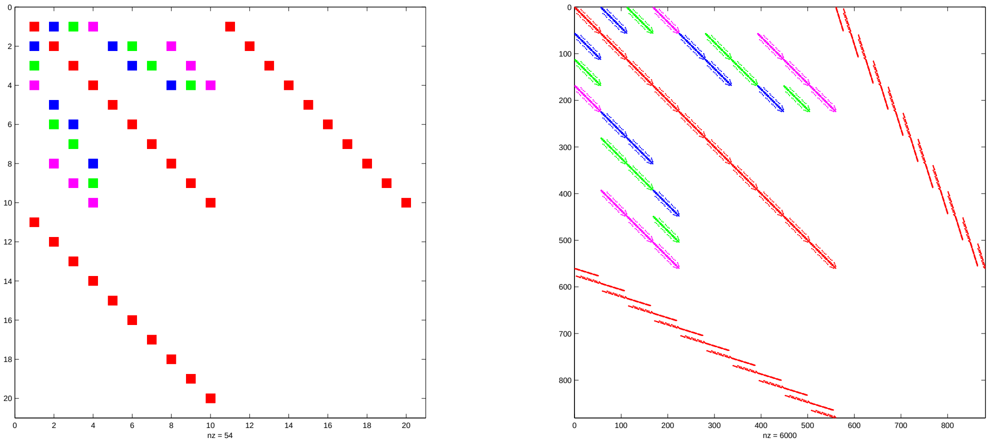
The potential $u^h$ and flux (or velocity) $\mathbf{q}^h$ are expressed in terms of the expansions

$$u^h(\mathbf{x},\xi) = \sum_{s=1}^{P}\sum_{r=1}^{N_e} u_{s,r}\phi_r(\mathbf{x})\chi_s(\xi), \qquad \mathbf{q}^h(\mathbf{x},\xi) = \sum_{s=1}^{P}\sum_{r=1}^{N_{edg}} \mathbf{q}_{s,r}\psi_r(\mathbf{x})\chi_s(\xi). \tag{4.46}$$

Substituting for $u^h$ and $\mathbf{q}^h$ using expansions (4.46) into (4.45), we obtain the discrete linear system

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix}, \tag{4.47}$$

where $A$ is a sparse symmetric matrix of size $N_{edg}P \times N_{edg}P$ with block structure and $B$ is an unsymmetric sparse matrix of size $N_eP \times N_{edg}P$ with block diagonal structure. For the example in which $p = 3$ and $d = 2$, the block structure of $C$, where $C = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$ is illustrated in Figure 4.2a.



(a) Block structure of $C$

(b) Sparsity pattern of $C$

Figure 4.2: (a) Block structure and (b) sparsity pattern of the global stiffness matrix $C$ - first order problem SMFEM

The structure of $C$ is governed by the coefficients of the polynomial chaos expansion (see Figure 4.6.1), whilst the sparsity of each of the blocks of $C$ (see Figure 4.2b) corresponds to the sparsity of the deterministic velocity matrix and the deterministic divergence operator. For the case of $h = \frac{1}{4}$ the sparsity of $C$ is illustrated in Figure 4.2b.

The diagonal blocks $A^*$ are products of the mean velocity matrix, $K_0$, and $\langle \chi_i^2 \rangle$

$$A^*_{i,i} = \langle \chi_i \rangle^2 \otimes K_0, \tag{4.48}$$

where

$$(K_0)_{r,s} = \int_D \frac{1}{\mu} \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}. \tag{4.49}$$

The off-diagonal blocks $A^\star$ are given by

$$A^\star_{i,j} = \sum_{l=1}^{d} \left[ \langle \xi_l \chi_i \chi_j \rangle \right] \otimes K_l, \tag{4.50}$$

where

$$(K_l)_{r,s} = \sigma \sqrt{\lambda_l} \int_D \beta_l(\mathbf{x}) \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}. \tag{4.51}$$

The block diagonal matrix $B$ is given by

$$B_{i,i} = \langle \chi_i \rangle^2 \otimes B_0, \tag{4.52}$$

where

$$B_0(r,s) = \int_D \phi_r(\mathbf{x}) \nabla \cdot \psi_s(\mathbf{x}) d(\mathbf{x}). \tag{4.53}$$

When the conductivity coefficient is approximated by (4.8), $A$ reduces to size $N_{edg}p \times N_{edg}p$ and has a tridiagonal structure and $B$ reduces to size $N_e p \times N_{edg}p$. Then, $C$ has the following structure

$$C = \begin{bmatrix} A_1^* & A_2^\star & & & & B_1^T & & & & \\ A_2^\star & A_2^* & A_3^\star & & & & B_2^T & & & \\ & \ddots & \ddots & \ddots & & & & \ddots & & \\ & & A_{p-1}^\star & A_{p-1}^* & A_p^\star & & & & B_{p-1}^T & \\ & & & A_p^\star & A_p^* & & & & & B_p^T \\ B_1 & & & & & & & & & \\ & B_2 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & B_{p-1} & & & & & & \\ & & & & B_p & & & & & \end{bmatrix}. \tag{4.54}$$

The diagonal blocks $A^*$ are given by (4.34) and the off-diagonal blocks $A^\star$ have the form

$$A^\star = \langle k_k \chi_i^k \chi_j^k \rangle \otimes K, \tag{4.55}$$

where

$$K(r,s) = \int_D \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d(\mathbf{x}), \tag{4.56}$$

and $k$ is as defined in (4.8).

### 4.6.2 Implementation and Solution Strategies

As for the SFEM method the coefficient matrix $C = \left[\begin{smallmatrix} A & B^T \\ B & 0 \end{smallmatrix}\right]$ is never assembled. In addition to storing $d+1$ matrices of size $N_{edg} \times N_{edg}$ and the entries of the stochastic matrices $G_k$, we also store the matrix $B_0$ of size $N_e \times N_{edg}$.

The efficient solution of the saddle-point system (4.47) is an active field of research (see Furnival (2008), Ernst et al. (2009) and Elman et al. (2010)). Our approach follows from our understanding of the deterministic system (see §2.5.3). In Chapter 3 we used MINRES equipped with a practical preconditioner based on the approximation of the Schur complement by sparse direct or algebraic multigrid methods.

For the stochastic system we use a preconditioner which follows from its deterministic version (2.41) and it is given by

$$\mathcal{P} = \begin{pmatrix} \tilde{N} & 0 \\ 0 & \tilde{V} \end{pmatrix}, \tag{4.57}$$

where

$$\tilde{N} = diag(A) = G_0 \otimes diag(K_0), \tag{4.58}$$

and

$$\tilde{V} = B\tilde{N}^{-1}B^T = G_0 \otimes \left[B_0 diag(K_0)^{-1} B_0^T\right]. \tag{4.59}$$

Following the discussion for the second order problem carried out in §4.5.3, the preconditioner (4.57) is expected to be efficient only for problems in which the spatial random field is characterised by small or moderate standard deviation. The fact is that, to date, a robust preconditioner in terms of $\sigma$, for the mixed stochastic formulation has not yet been proposed.

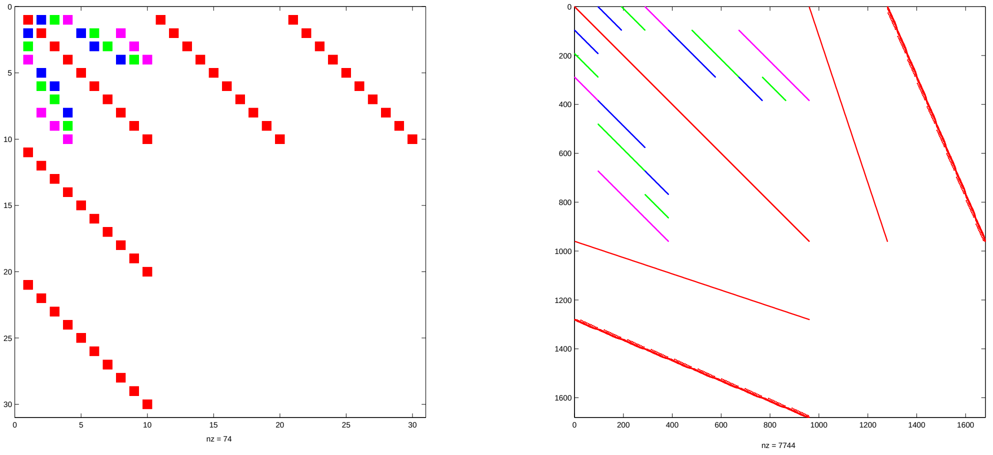
Due to the structure of $C$, the new developments reported in §4.5.3 are not equally applicable to the first order problem. The stochastic version of the mixed-hybrid method could provide the solution to this drawback. This is discussed further in the next sections.

## 4.7 Stochastic Mixed Hybrid Formulation

We have seen that, in the deterministic case, the indefinite discrete linear system obtained by mixed methods can be reduced to a positive definite system of equations. The success of the hybridization technique relies on the fact that the matrix $A$ is diagonal, hence the computation of $A^{-1}$ is cheap. Furthermore, the Schur complement $BA^{-1}B^T$ and its inverse are also diagonal and efficiently computed. The resulting matrix $D$ is sparse and positive definite.

The stochastic global system of the mixed hybrid method has a similar form to the one associated with the deterministic counterpart (2.49). However, the global matrix $A$ is now a block matrix whose sparsity is governed by the coefficients of the polynomial chaos expansion. Note that each block of $A$ is diagonal. Its block structure is shown in Figure 4.3(a) for the case of $d = 3$ and $p = 2$. Its sparsity pattern is shown in Figure 4.3(b).

The global stochastic matrices $B = \langle \chi_i \rangle^2 \otimes B_0$ and $C = \langle \chi_i \rangle^2 \otimes C_0$, $i = 1, \ldots, P$, where $B_0$ and $C_0$ are the corresponding deterministic global matrices, are block diagonal. Their block structure and sparsity pattern is also shown in Figure 4.3(a) and Figure 4.3(b).



(a) Block structure

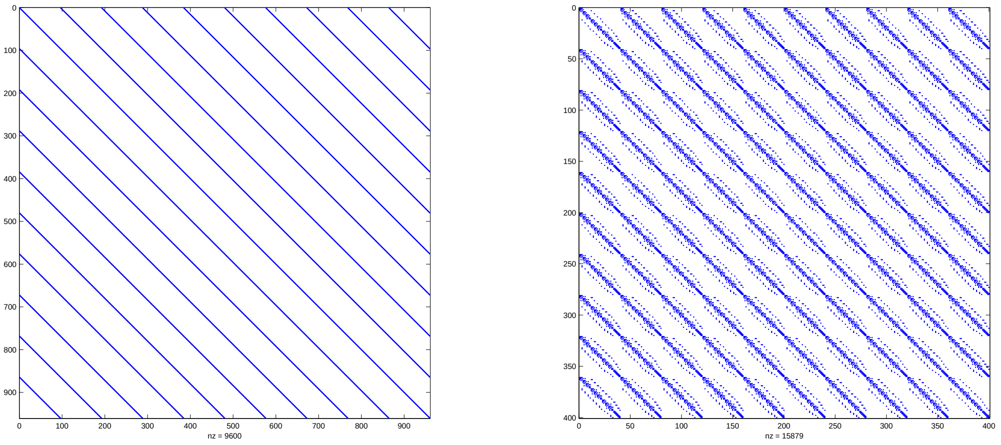
(b) Sparsity pattern

Figure 4.3: (a) Block structure and (b) sparsity pattern of the global stiffness matrix for $d = 3$ and $p = 2$ - first order problem SMHFEM

The global matrix $A$ is sparse and posses a block structure but, contrary to the deterministic version, has diagonal entries for every non-zero entries of the stochastic matrices $G_k$, $k = 0, ..., d$ in 4.38. Several blocks of $A$ are zero reflecting the sparsity of the polynomial chaos coefficients. However, the inverse of $A$ is diagonally dominant (having diagonal entries for every block) as shown in Figure 4.4a. It becomes apparent that the computation of $A^{-1}$ is still feasible but it is less computationally straightforward than for the deterministic case.

In addition to the computation of the inverse of $A$, the inverse of the Schur complement (2.54) needs to be performed. The operation $BA^{-1}B^T$ and its inverse produce matrices exhibiting the same sparsity pattern as $A^{-1}$ (see Figure 4.4a).

Given the structure and sparsity patterns of the inverse matrices, the global stochastic matrix $D$ (obtained using the post-processing technique (2.54)) has entries for each block. Its sparsity pattern is illustrated in Figure 4.7. Each block has



(a) Sparsity pattern of $A^{-1}$

(b) Sparsity pattern of $D$

Figure 4.4: Sparsity patterns for $A^{-1}$ and $D$

non-zero entries corresponding to the sparsity pattern of the deterministic matrix $D$.

### 4.7.1 Implementation and Solution Strategies

Cases for which the SMHFEM has been implemented are not available in the literature. In fact, it appears that most of effort has concentrated on the efficient solution of SMFEM and, as far as I know, the stochastic mixed-hybrid implementation is yet to be explored.

The attractiveness of this method is due to the fact that the global matrix $D$, although having non-zero contributions for each block, has a structure similar to the second order problem. Consequently the preconditioners which are proposed in §4.5.3 can be used to solve this system of equations efficiently.

However, there are some very important matters to consider. Firstly, the effective applicability of the method depends on the efficient inversion of $A$ and $BA^{-1}B^T$. This could be achieved using block Gaussian elimination. In fact, only multiplications of diagonal matrices are required and this should be very fast and efficient. Secondly, it appears that all the blocks of the coefficient matrix need to be stored in contrast to the other implementations. In the SFEM and SMFEM implementations only $d+1$ matrices and the entries of the stochastic matrices $G_k$ need to be stored. It seems that the same implementation is not possible for the SMHFEM, due to the post-processing with which $D$ is ultimately obtained. The memory required to store a large number of matrices represents a significant limitation of this method.

In this thesis we do not present numerical simulations for the stochastic version of the mixed-hybrid finite element method. The two challenges mentioned above have to be evaluated carefully to understand if the implementation of this method is feasible. This topic requires further research.

# Chapter 5

# A Comparison of Stochastic Galerkin and Monte Carlo Methods

## 5.1 Introduction

The aim of this chapter is to compare the solutions obtained by means of Stochastic Galerkin methods (SFEM / SMFEM) with those obtained by traditional Monte Carlo methods (MCM). The comparison of solutions give us the possibility to validate the SG numerical development. For the Monte Carlo simulations we use the deterministic mixed method which was validated in Chapter 3 using test problems with known analytical solutions. Therefore, the statistics obtained by sequential deterministic simulations (MCM) represent a suitable data-set against which to validate the results obtained with SG methods.

It should be noted that the numerical solutions for the test problems presented in this chapter (see §5.2 and §5.3) using SG and MCM do not 'exactly' converge to the results but some (small) differences are expected. The primary objective is to show that the numerical examples produce physically meaningful and comparable results.

In addition to validation purposes the chapter also includes a convergence analysis for the two methods. Specifically, we consider the methods to have converged if there is no significant change in the solution for progressively larger polynomial orders (SFEM / SMFEM cases) and number of simulations (MCM case). Ideally, the methods are considered to have converged if the first four significant digits of the solutions do not vary with increasing polynomial order $p$ and number of simulations $N_r$. Although, this analysis could be considered subjective, it is our aim to highlight the different convergence rates of the two methodologies.

For each of the test problems presented in this chapter the performance associated with both methodologies is reported. Although a robust comparison of the computational cost required by the two methods is feasible, this is outside the scope of this thesis. In fact the numerical codes implemented in this work are prototypes and still under development. A formal comparison between MCM and SG requires state-of-the-art algorithms and solvers and therefore this is an objective for future work. The timings listed in the tables should only give the reader an indication of the CPU cost required by that specific method. The simulations have all been carried out in serial within the MATLAB 7.4 on a laptop PC with $4Gb$ of RAM. A comparison of CPU performance for the numerical experiments based on parallel implementation algorithms is not considered in this thesis.

Additionally, it should be borne in mind that for the MCM the time required to create $N_r$ spatial random field realizations can be very large and in some circumstances greater than the actual solution time. However, it is recognized that this computational time depends largely on the algorithm and method chosen to discretize the random field. Therefore, this CPU cost is omitted from the MCM timings reported in the tables. The timings reported for the SG simulations are total times and they also include the CPU cost required to discretise the spatial random fields.

Two test problems are considered in this chapter aimed at validating the SG implementation based on Gaussian and uniform distributions. The test problems also address the methods convergence behaviour for various settings. In particular we look at cases with mixed boundary conditions and small and large variances. A comparison of MCM and SG methods for test problems in which the conductivity coefficient is modelled by lognormal distributions is reported in Chapter 7.

The author would like to acknowledge the use of some of the MATLAB-based functions made available by Sudret & Der Kiureghian (2000) for the experiments carried out in this chapter.

## 5.2 SFEM vs Monte Carlo Simulations

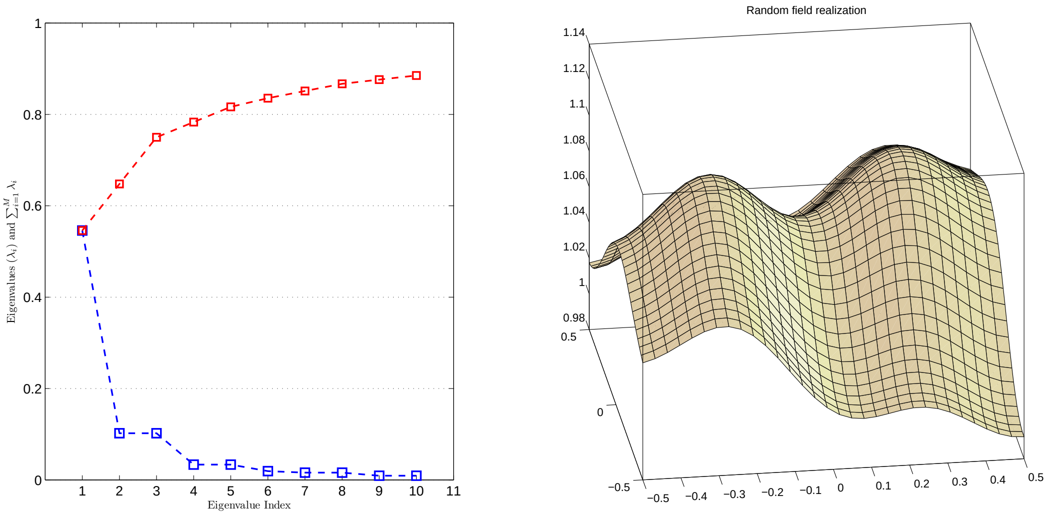
For the SFEM the conjugate gradient method, preconditioned with $P_{\lfloor \mathcal{SGS}}$ based on (*UMFPACK*), is used (see §4.5.3). For the MC method a sparse direct solver (*UMFPACK*) is used. This is a suitable choice given that the mesh used in the following experiments is relatively coarse ($h = \frac{1}{32}$).

### 5.2.1 Test Problem 1 - Hermite polynomials

This test problem is taken from Deb et al. (2001). We consider the square domain $[-0.5, 0.5] \times [-0.5, 0.5]$ and source term $f(x, y) = 2(0.5 - x^2 - y^2)$. The stochastic system of equations to be solved is given in (4.1) with homogeneous Dirichlet boundary conditions defined everywhere in $\Gamma$. The problem is solved on a regular triangular mesh with discretization parameter $h = \frac{1}{32}$.

The spatial variability of the conductivity coefficient $\mathcal{C}^h(\mathbf{x}, \xi(\omega))$ is described by an exponential correlation function in which the correlation lengths are $l_x = l_y = 1.0$. The spatial random field is assumed to be normally distributed with $\mu = 1$ and

$\sigma = 0.1$. The eigenvalues and eigenfunctions of the Karhunen-Loève expansion of $\mathcal{C}$ are available as analytical expressions (Ghanem & Spanos 2003, Powell & Elman 2009). These can be expressed as the products of those of two corresponding 1D problems. Closed form solutions to the eigenvalue problems are given in Ghanem & Spanos (2003). Note that closed form solutions are only available for the cases in which the covariance function is exponential / square exponential or triangular and for squared or rectangular domains. Figure 5.1a shows the decay of the first 10 eigenvalues obtained from the KLE as well as their summation. Figure 5.1b illustrates a realization of the conductivity field for test problem 1.



(a) KLE eigenvalues for exponential covariance and $l_x = l_y = 1$

(b) Sample realization of spatial random field

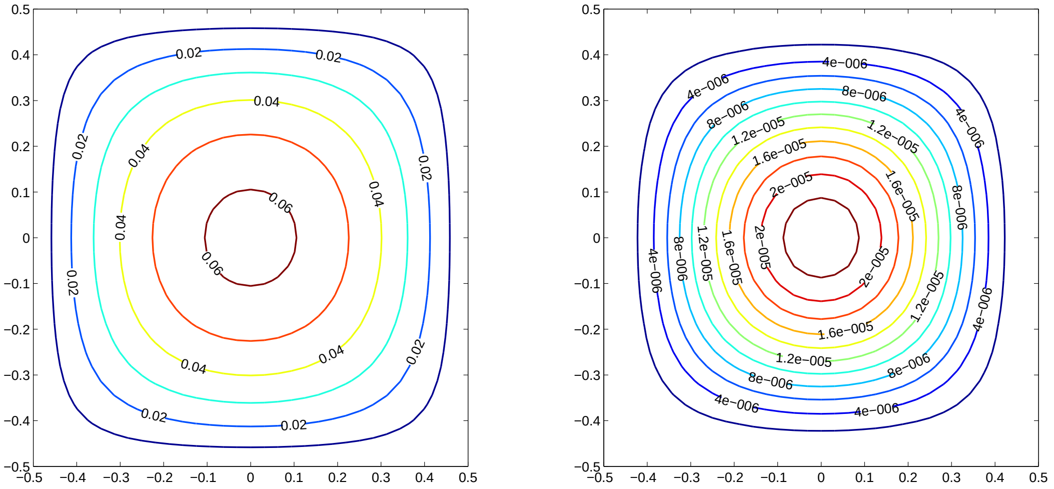
Figure 5.1: KLE eigenvalues and sample realization of $\mathcal{C}(\mathbf{x}, \xi)$ for test problem 1

Normally distributed random variables are used in this test problem. Hence the basis functions for the stochastic space are $d$-variate Hermite polynomials of order less than or equal to $p$. The spatial domain is discretised by a triangulation consisting of a 32×32 grid of squares each of which is further divided into two triangles. Thus, the size of the deterministic problem corresponds to the number of nodes, $N_u = 1,089$. Table 5.1 reports the overall number of equations solved using MCM and the dimension of the stochastic space and global stiffness matrix obtained using SFEM.

Table 5.1: Dimension of MCM and SFEM for test problem 1

| $N_r$ | $10,000$ | $20,000$ | $40,000$ |
|---|---|---|---|
| $MCM\#eq.$ | $10,890,000$ | $21,780,000$ | $43,560,000$ |
| $p$ | 2 | 3 | 4 |
| $P$ | 28 | 84 | 210 |
| $SFEM\#eq.$ | $30,492$ | $91,476$ | $228,690$ |

The mean and variance solutions for the potential obtained using SFEM (with $d = 6$ and $p = 4$) on a $32 \times 32$ uniform grid are shown in Figure 5.2.

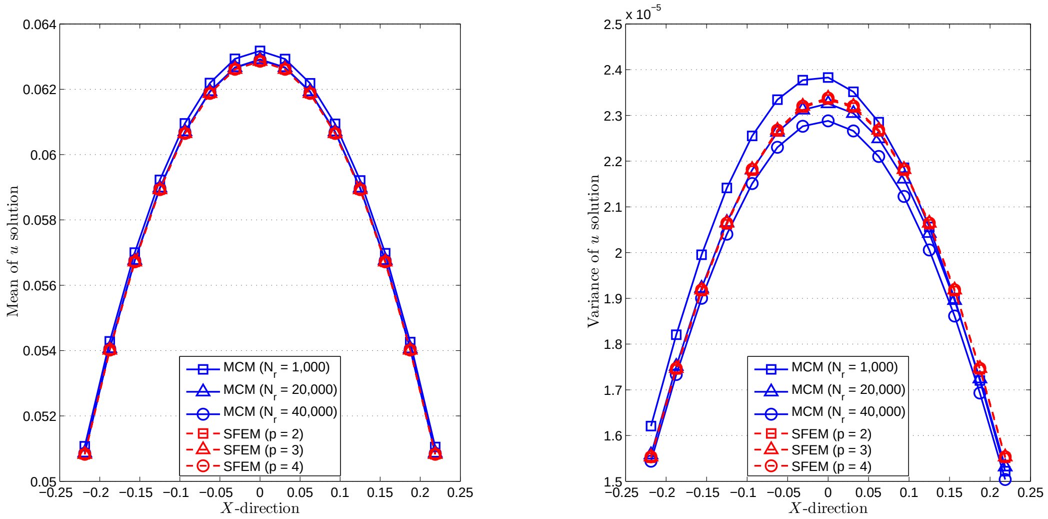


(a) Mean $u$ solution

(b) Variance $u$ solution

Figure 5.2: SFEM mean and variance solutions for the potential, $u(x, y)$, for test problem 1

Figure 5.3 shows the mean and variance solution profiles along the horizontal centreline of the domain for several values of polynomial order $p$ and number of MC simulations $N_r$. Note that Figure 5.3 presents data only for the interval $[-0.25 < X < 0.25]$, which is the part of the domain furthest away from the boundaries.



(a) Mean $u$ solution

(b) Variance $u$ solution

Figure 5.3: Comparison of solution profiles for SFEM and MCM for test problem 1

Table 5.2 shows the value of the mean and variance at location $(0.5, 0.5)$ for several values of $N_r$ and $p$. The SFEM converges rapidly to the desired solution. Hermite polynomials of second-order ($p = 2$) are sufficient to achieve convergence to the fourth significant digit for the mean solution. Polynomials of order three are required to obtain the same level of accuracy for the variance solution.

Although the spatial random field is characterized by a small variance, the Monte Carlo method converges slowly. Table 5.2 shows that $20,000$ simulations are sufficient to achieve convergence to the third significant digit for the mean solution. However the variance solution requires $30,000$ simulations (sample variance $0.0000227\underline{6}2$) to converge to the second significant digit. Note also that in contrast to the SFEM solution the MC variance solution is not symmetric (see Figure 5.2).

Table 5.2 also reports CPU times required to solve one single discrete linear system (SFEM) and $N_r$ deterministic problems (MCM). The CPU timings indicate that for this test problem SFEM is significantly more efficient than MCM. However, it

Table 5.2: Convergence analysis of MCM and SFEM for test problem 1

| | $N_r = 10,000$ | $20,000$ | $N_r = 40,000$ |
|---|---|---|---|
| Sample Mean | $0.0629\underline{1}5$ | $0.0629\underline{2}8$ | $0.0629\underline{0}4$ |
| Sample Variance | $0.000023\underline{2}90$ | $0.000023\underline{2}57$ | $0.000022\underline{8}81$ |
| $t_{CPU}(sec.)$ | 15.82 | 31.64 | 63.29 |
| | $p = 2$ | $p = 3$ | $p = 4$ |
| Mean | $0.0628\underline{5}5$ | $0.0628\underline{5}6$ | $0.0628\underline{5}6$ |
| Variance | $0.00002333\underline{4}8$ | $0.00002333\underline{7}7$ | $0.00002333\underline{7}8$ |
| $t_{CPU}(sec.)$ | 0.39 | 1.54 | 3.96 |

should be kept in mind that more efficient algorithms could be developed for both methodologies, providing different time estimates. Therefore, the timings reported serve as an indication of efficiency only and do not provide a robust comparison between the two methodologies. Furthermore, it should be noted that the linear growth in CPU time reported for the MCM is not observed for the SFEM. In fact, for the latter method the dimension of the stochastic discrete linear system increases factorially with the maximum order, $p$, of the polynomials used to discretise the stochastic space.

### 5.2.2 Test Problem 2 - Legendre polynomials

The second test problem is similar to the one presented in Powell & Elman (2009). We consider the square domain $[0.0, 1.0] \times [0.0, 1.0]$ with source term $f(x, y) = 1$. Dirichlet boundary conditions are imposed on the left and right edge of the square domain such that $\Gamma_D = \{0, 1\} \times [0, 1]$. Homogeneous Neumann boundary conditions are imposed on the upper and lower edge of the domain so that the flow is tangent to these boundaries. The system of equation defined in (4.1) is solved on a regular triangular mesh with discretization parameter $h = \frac{1}{32}$.

The spatial model for $\mathcal{C}^h(\mathbf{x}, \xi(\omega))$ is the same as the one described for test problem 1. However for this test problem we set the standard deviation to $\sigma = 0.7$.

In this test problem random normal variables cannot be employed because when a large standard deviation is deployed the discrete linear system and preconditioned system become indefinite. In fact it can be shown that when Hermite polynomials are used, the positive definiteness of the coefficient matrix is never guaranteed. (Powell & Elman 2009) showed that when Hermite polynomials are used, for fixed values of $h$, $d$ and $\sigma$ there is always a value of $p$ that determines the coefficient matrix $A$ to be indefinite. Furthermore the author showed that for small values of $\sigma$ the link between $A$ being SPD and the order of the Hermite polynomials $p$ is not evident.

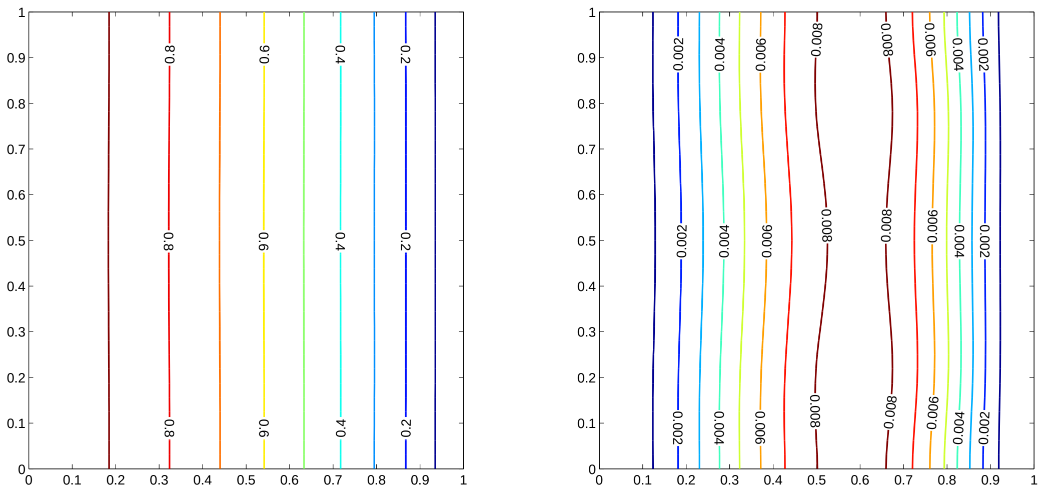
Thus, given that in this test problem $\sigma$ is large, normal distributions cannot be used. Therefore, independent and uniformly distributed random variables, defined in the interval $(-1, 1)$, are used. Hence the basis functions for the stochastic space are $d$-variate Legendre polynomials of order less than or equal to $p$. Let us set $d = 4$ (four random variables) and use polynomials up to order eight. The spatial domain is discretised by the same triangulation described in test problem 2. Table 5.3 reports the overall number of equations solved using MCM and the dimension of the stochastic space and global stiffness matrix obtained using SFEM.

Table 5.3: Dimension of MCM and SFEM for test problem 2

| $N_r$ | $10,000$ | $20,000$ | $40,000$ | $80,000$ |
|---|---|---|---|---|
| $MCM\#eq.$ | $10,890,000$ | $21,780,000$ | $43,560,000$ | $87,120,000$ |
| $p$ | 5 | 6 | 7 | 8 |
| $P$ | 126 | 210 | 330 | 495 |
| $SFEM\#eq.$ | $137,214$ | $228,690$ | $359,370$ | $539,055$ |

The mean and variance solutions for the potential obtained using SFEM with $p = 8$ and $d = 4$ on a $32 \times 32$ uniform grid are illustrated in Figure 5.4.

Figure 5.5 shows the mean and variance solution profiles along the horizontal centreline of the domain for several values of polynomial order $p$ and number of MC simulations $N_r$. As for test problem 1 the solution profiles obtained by the two

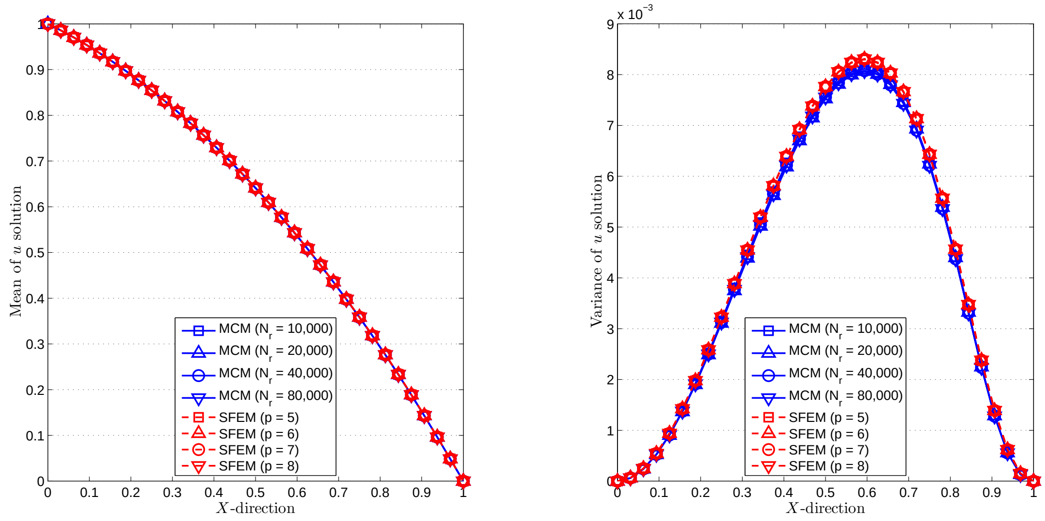


(a) Mean $u$ solution

(b) Variance $u$ solution

Figure 5.4: Mean and variance solutions for the potential for test problem 2

methods are very similar and converge to the same values for increasing sampling size, $N_r$, and polynomial order, $p$.



(a) Mean $u$ solution

(b) Variance $u$ solution

Figure 5.5: Comparison of solution profiles for SFEM and MCM for test problem 2

Table 5.4 shows the value of the mean and variance at location $(0.5, 0.5)$ for several values of $N_r$ and $p$. Legendre polynomials of order four (not shown in Table 5.4) are sufficient to achieve convergence to the fourth significant digit for the mean solution. Polynomials of order seven, instead, are required for the variance solution to achieve the same level of accuracy. Hence, not surprisingly, the first moment solution always converges more rapidly than the second moment solution, no matter how large the variability of the spatial random field is.

For random fields with large variance the Monte Carlo method converges very slowly. Table 5.4 shows that $40,000$ simulations are required for the sample mean to converge. Furthermore, the maximum size of the sample used in this study ($N_r = 80,000$) is not sufficient to achieve convergence for the variance solution.

Table 5.4: Convergence analysis of MCM and SFEM for test problem 2

| | $N_r = 10,000$ | $N_r = 20,000$ | $N_r = 40,000$ | $N_r = 80,000$ |
|---|---|---|---|---|
| Sample Mean | $0.641\underline{3}2$ | $0.640\underline{8}7$ | $0.640\underline{7}2$ | $0.640\underline{7}0$ |
| Sample Variance | $0.00759\underline{6}4$ | $0.00752\underline{4}4$ | $0.00755\underline{5}6$ | $0.007530\underline{0}$ |
| $t_{CPU}(sec.)$ | 25.70 | 51.40 | 102.80 | 205.61 |
| | $p = 5$ | $p = 6$ | $p = 7$ | $p = 8$ |
| Mean | $0.641\underline{1}4$ | $0.641\underline{1}5$ | $0.641\underline{1}5$ | $0.641\underline{1}5$ |
| Variance | $0.00775\underline{8}0$ | $0.00776\underline{3}9$ | $0.00776\underline{5}0$ | $0.00776\underline{5}3$ |
| $t_{CPU}(sec.)$ | 5.40 | 13.26 | 19.78 | 21.54 |

The CPU times reported in Table 5.4 indicate that the SFEM method is significantly more efficient than the MCM when large standard deviations and Legendre polynomials are used. Although the results presented in test problems 1 and 2 are not directly comparable, SFEM using Legendre polynomials is generally more efficient than using Hermite polynomials. Furthermore, in the latter case the positive definiteness of the coefficient matrix is guaranteed only when the standard deviation and polynomial order are not too large. Thus it would not be possible to obtain a solution for test problem 2 if Hermite polynomials were used.

## 5.3 SMFEM vs Monte Carlo Simulations

For the SMFEM and for each MC simulation, MINRES equipped with a Schur complement preconditioner based on AMG is used (see §2.5.3 and §4.6.2). For the deterministic case this choice is motivated by the outcomes of Chapter 3. For the stochastic problem instead this is the most efficient and practical preconditioner currently available.

### 5.3.1 Test Problem 1 - Hermite polynomials

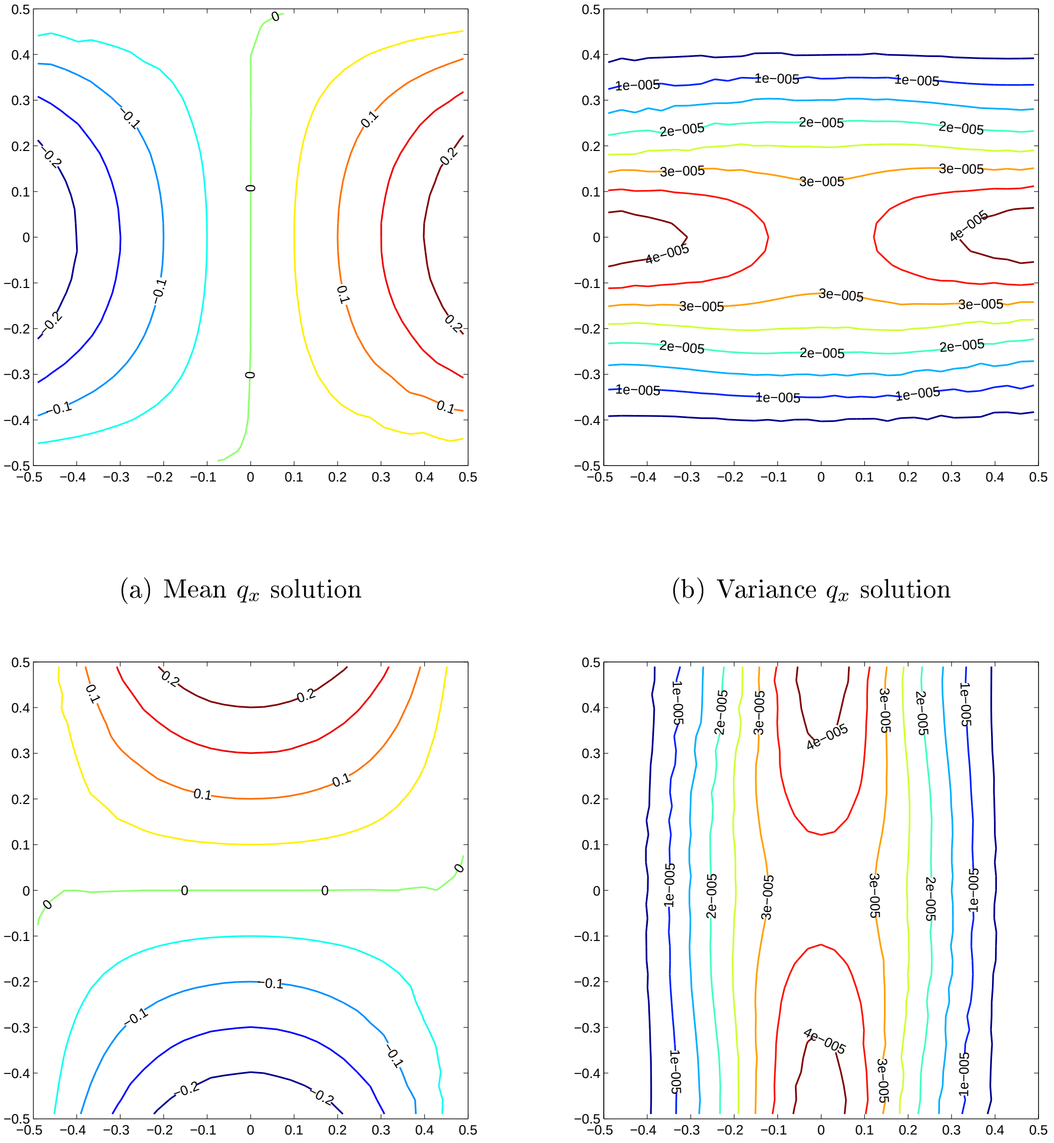
The settings for this test problem are described in §5.2.1.

The solution of the stochastic mixed formulation provides, in addition to the mean and variance of the potential, the mean and variance of the two components of the velocity field. In fact, as explained in Chapter 2, simultaneous solutions are obtained for the potential, at the centroid of the finite elements and for the normal fluxes at the edges of the triangulation. Thus, the size of the deterministic problem corresponds to the sum of the number of elements, $N_e = 2,048$, and number of edges, $N_{edg} = 3,136$. Table 5.5 reports the overall number of equations solved using MCM and the dimension of the stochastic space and global stiffness matrix obtained using SMFEM.

Table 5.5: Dimension of MCM and SMFEM for test problem 1

| $N_r$ | $10,000$ | $20,000$ | $40,000$ |
|---|---|---|---|
| $MCM\#eq.$ | $51,840,000$ | $103,680,000$ | $207,360,000$ |
| $p$ | 2 | 3 | 4 |
| $P$ | 28 | 84 | 210 |
| $SFEM\#eq.$ | $145,512$ | $435,456$ | $1,088,640$ |

The mean and variance for the potential solution are similar to those obtained using SFEM and they are illustrated in Figure 5.1. The mean and variance solutions for the two components of the velocity field obtained using SMFEM (with $d = 6$ and $p = 4$) on a $32 \times 32$ uniform grid are shown in Figure 5.6.



(a) Mean $q_x$ solution

(b) Variance $q_x$ solution

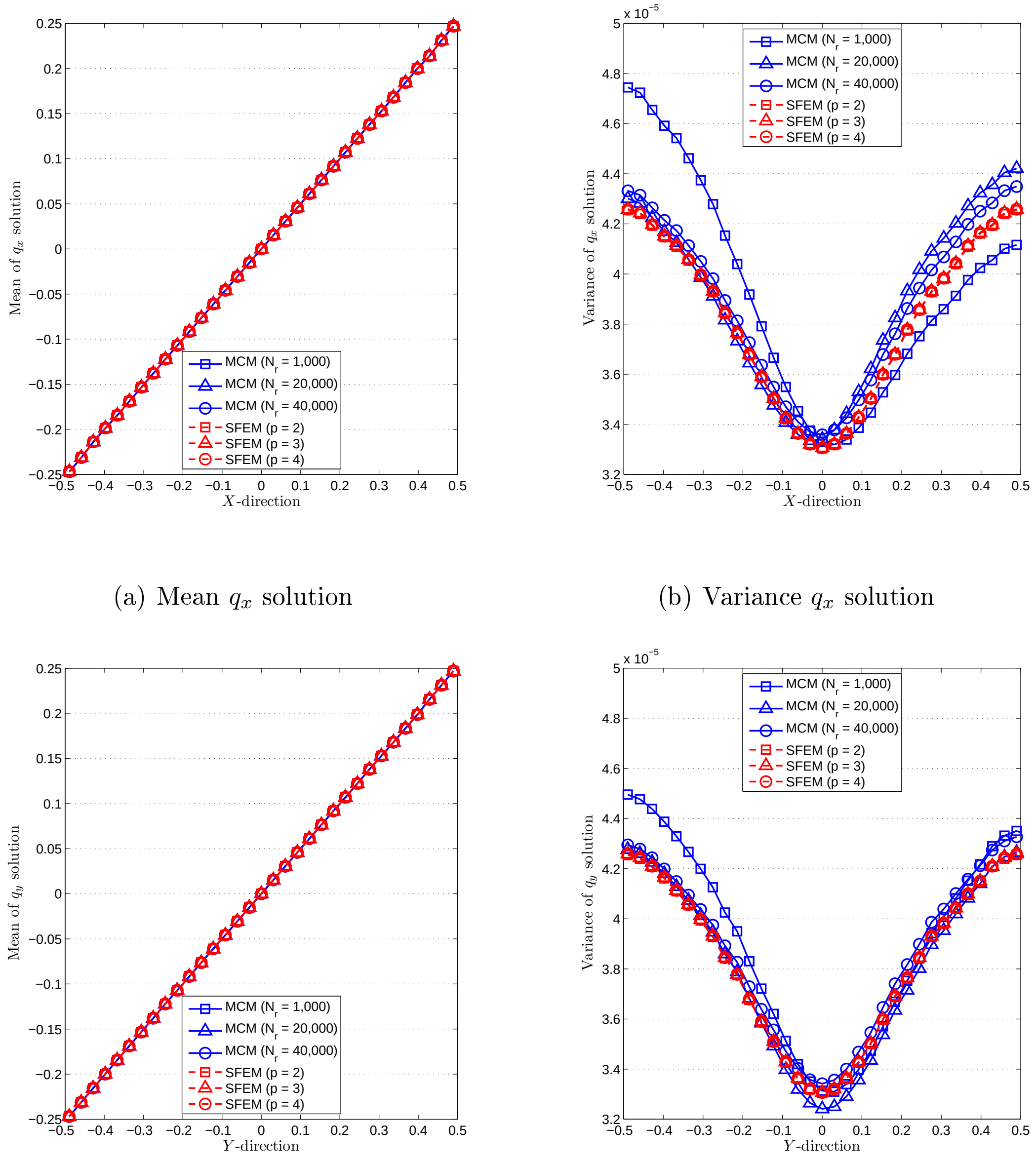
(c) Mean $q_y$ solution

(d) Variance $q_y$ solution

Figure 5.6: Mean and variance solutions for test problem 1 for $h = \frac{1}{32}$

The mean and variance profiles for the $X$ component of the velocity field along the horizontal centreline of the domain for several values of polynomial orders $p$ and number of MC simulations $N_r$ are illustrated in Figure 5.8. The profiles for the $Y$

component along the vertical centreline are also illustrated in the same figure. As for the second order problem the mean and variance of the velocity field for the MCM and SMFEM converge to the same solution for increasing sampling size, $N_r$, and polynomial order, $p$.



(a) Mean $q_x$ solution (b) Variance $q_x$ solution

(c) Mean $q_y$ solution (d) Variance $q_y$ solution

Figure 5.7: Solution profiles along the horizontal and vertical centerline

Table 5.6 shows the values of the mean and variance solutions sampled at locations

$(-0.5, 0.0)$ and $(0.0, -0.5)$ for several values of $N_r$ and $p$, for the $X$-component and $Y$-component of the velocity field, respectively.

The SMFEM converges at the same rates reported for the SFEM case, i.e. polynomials of order two are sufficient for the mean velocity solutions and polynomials of order three are required for the variance velocity solutions to achieve accuracy to the fourth significant digit. Note also that the $q_x$ and $q_y$ solutions are perfectly symmetric.

The Monte Carlo mean solution of the velocity field converges rapidly and it appears that $10,000$ simulations are sufficient to achieve convergence. However, the variance solution for the velocity components converges slowly and it appears to be not symmetric (see Table 5.6 and Figure 5.5). Consequently, the rate of convergence for the $X$ and $Y$ velocity components differ, slightly. Results presented in Table 5.6 indicate that $20,000$ simulations are sufficient to achieve convergence to the third significant digit for $\mathbf{q}_y$. However, $40,000$ simulations or more are required to achieve the same level of accuracy for $\mathbf{q}_x$.

Table 5.6: Convergence analysis of MCM and SMFEM for test problem 1

| | | $N_r = 10,000$ | $N_r = 20,000$ | $N_r = 40,000$ |
|---|---|---|---|---|
| $\mathbf{q}_x$ | Sample Mean | $-0.246\underline{9}4$ | $-0.246\underline{9}9$ | $-0.246\underline{9}2$ |
| | Sample Variance | $0.000043\underline{0}08$ → $0.0000430\underline{0}8$ | $0.0000435\underline{6}7$ | $0.0000433\underline{2}0$ |
| $\mathbf{q}_y$ | Sample Mean | $-0.247\underline{0}6$ | $-0.246\underline{9}6$ | $-0.246\underline{9}4$ |
| | Sample Variance | $0.0000427\underline{7}0$ | $0.0000429\underline{8}2$ | $0.0000429\underline{4}7$ |
| | $t_{CPU}(sec.)$ | $2,023$ | $4,046$ | $8,092$ |
| | | $p = 2$ | $p = 3$ | $p = 4$ |
| $\mathbf{q}_x$ | Mean | $-0.246\underline{8}8$ | $-0.246\underline{8}8$ | $-0.246\underline{8}8$ |
| | Variance | $0.0000425\underline{4}7$ | $0.0000425\underline{8}4$ | $0.0000425\underline{8}5$ |
| $\mathbf{q}_y$ | Mean | $-0.246\underline{8}8$ | $-0.246\underline{8}8$ | $-0.246\underline{8}8$ |
| | Variance | $0.0000425\underline{4}7$ | $0.0000425\underline{8}4$ | $0.0000425\underline{8}5$ |
| | $t_{CPU}(sec.)$ | 11.02 | 44.77 | 160.82 |

The CPU cost per simulation is significantly more expensive for the first than for the second order problem. In a stochastic context, where several thousand simulations are required, the MC method becomes computationally very expensive. This is clear from Table 5.6, where the reported data show that more than two hours are required to solve 40,000 linear systems of equations on a relatively coarse grid ($h = \frac{1}{32}$). It should be noted that in real life applications the size of the sample will be significantly larger than the one considered in this test problem.

On the other hand, the SMFEM is significantly more efficient. Note, however, that this conclusion cannot be generalized as the random field used in this test problem has a low standard deviation. The next example shows that the SMFEM solution time increases significantly for problems with larger standard deviations.

### 5.3.2 Test Problem 2 - variable $\sigma$

The settings for this test problem are described in §6.2.2.

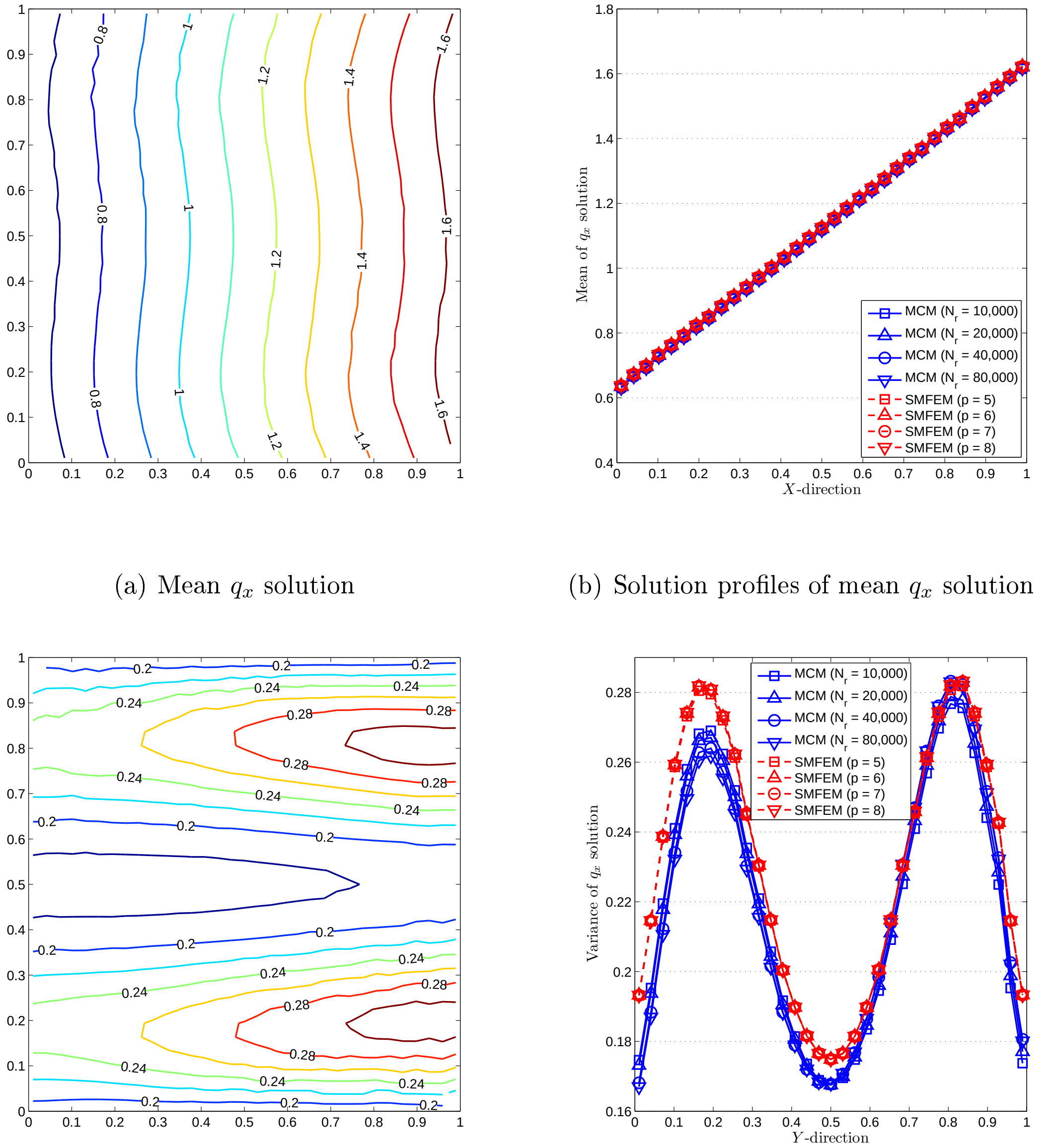
Table 5.7 reports the overall number of equations solved using the MC method for test problem 2. The table also includes the dimension of the stochastic space and the global stiffness matrix obtained using SMFEM.

Table 5.7: Dimension of MCM and SMFEM for test problem 2

| $N_r$ | $10,000$ | $20,000$ | $40,000$ | $80,000$ |
|---|---|---|---|---|
| $MCM\#eq.$ | $51,840,000$ | $103,680,000$ | $207,360,000$ | $414,720,000$ |
| $p$ | $5$ | $6$ | $7$ | $8$ |
| $P$ | $126$ | $210$ | $330$ | $495$ |
| $SFEM\#eq.$ | $653,184$ | $1,088,640$ | $1,710,720$ | $2,566,080$ |

The mean and variance solution for the potential are very similar to those obtained with the second order problem and these are illustrated in Figure 5.4. The mean and variance solutions for the components of the velocity field for $d = 4$ and $p = 8$ on a $32 \times 32$ uniform grid are shown in Figure 5.8. Note that for this test problem the flow is predominantly from left to right, hence the $Y$-component of the velocity field is equal or close to zero and therefore it is omitted from Figure 5.8.

Figure 5.8 also includes the solution profiles for various order of Legendre polynomials $p$ and various Monte Carlo samples, $N_r$. For the mean velocity ($X$-component) solution the profile presented is along the direction $Y = 0.5$ and for the variance solution is along the direction $X = 0.5$. An in-depth convergence study for a sampling point having coordinate $(0.5, 0.5)$ is reported in Table 5.8.



(a) Mean $q_x$ solution

(b) Solution profiles of mean $q_x$ solution

(c) Variance $q_x$ solution

(d) Solution profiles of variance $q_x$ solution

Figure 5.8: Mean and variance solutions for test problem 2 for $h = \frac{1}{32}$

Legendre polynomials of order four (not shown in Table 5.8) are sufficient to achieve convergence to the fourth significant digit for the mean solution. Polynomials of order six, instead, are required for the variance solution to achieve the same level of accuracy. This is in agreement with the convergence rate of the mean and variance solution for the potential recorded for the second order problem (see Table 5.4).

Noticeably it is apparent from the data presented in Table 5.8 that the Monte Carlo mean solution for the $X$ component of the velocity field does not converge for the sample size considered in this test problem. This is somewhat discordant if compared with the convergence rate of the potential solution for the first (not shown in Table 5.8) and second order problem (see Table 5.4). Equally the variance solution does not converge for the maximum sample size herein considered. However, convergence to the third significant digit is achieved for just $10,000$ simulations.

Table 5.8: Convergence analysis of MCM and SMFEM for test problem 2

| | | $N_r = 10,000$ | $N_r = 20,000$ | $N_r = 40,000$ | $N_r = 80,000$ |
|---|---|---|---|---|---|
| $\mathbf{q}_x$ | Sample Mean | $1.120\underline{8}3$ | $1.121\underline{8}2$ | $1.117\underline{9}9$ | $1.118\underline{3}5$ |
| | Sample Variance | $0.167\underline{5}5$ | $0.167\underline{5}6$ | $0.167\underline{7}7$ | $0.167\underline{9}5$ |
| | $t_{CPU}(sec.)$ | $1,437$ | $2,874$ | $5,748$ | $11,496$ |
| | | $p = 5$ | $p = 6$ | $p = 7$ | $p = 8$ |
| $\mathbf{q}_x$ | Mean | $1.125\underline{1}7$ | $1.125\underline{1}6$ | $1.125\underline{1}5$ | $1.125\underline{1}5$ |
| | Variance | $0.174\underline{9}1$ | $0.174\underline{8}8$ | $0.174\underline{8}9$ | $0.174\underline{8}9$ |
| | $t_{CPU}(sec.)$ | $374.67$ | $605.13$ | $1,649.94$ | $2,331.29$ |

The data on computational performance reported in Table 5.8 reveal that the performance of the solver used for the SMFEM deteriorates significantly for problems in which the spatial random field possesses a large standard deviation. This aspect is extensively covered in §4.6.2 and §4.5.3 and is the focus of the discussion in Chapter 6.

## 5.4 Conclusions

The primary objective of this chapter was to validate the numerical results obtained using Stochastic Galerkin methods with those obtained by traditional Monte Carlo methods.

We have shown that good agreement between the two methods has been achieved for problems in which the conductivity coefficient is described by Gaussian and uniform distributions. A similar analysis is carried out in Chapter 7 for problems with lognormal distributions. Other types of distributions such as Gamma or Beta are not considered in this thesis for they are less relevant for application in the groundwater modelling context.

In addition to our validation work, the chapter also reported an in-depth convergence analysis of SFEM/SMFEM and MCM. The main findings of this analysis are summarised as follows.

Generally, low order polynomials (up to fourth order for large standard deviation) are sufficient to achieve mean solution convergence to the fourth significant digit for Stochastic Galerkin methods. Conversely, the variance solution converges more slowly and higher order polynomials are generally required (up to seventh order for large standard deviation).

Monte Carlo methods show slow convergence rates even when the spatial random field is characterised by a small standard deviation. The numerical experiments suggest that for the variance solutions to converge a very large sample of realisations is generally required.

It is evident that the Monte Carlo method is computationally very expensive. In fact, for a specific problem the overall number of equations to be solved can be very large depending on the number of realisations considered. In comparison, the number of equations to be solved in a stochastic Galerkin implementation is typically just a fraction of that.

Nevertheless, the efficient use of SG methods is limited to problems where the conductivity coefficient is accurately approximated by a small number of random variables. This class of problems arises when the correlation lengths of the spatial model are of the same size as the physical domain or larger. In these circumstances the eigenvalues of the KLE decay rapidly, thus a small number of random variables are sufficient to accurately approximate the random field. For cases in which the correlation lengths of the spatial field are small and thus a large number of random variables need to be considered, the implementation of SG becomes impracticable.

Furthermore, we have shown that for problems in which the spatial random field is characterised by a large standard deviation, the performance of the solvers used for the SG methods deteriorates significantly. The preconditioners used are, in fact, not robust for this class of problems. It is therefore crucial that, in order for SG methods to be computationally competitive in all settings (small and large standard deviation), the chosen iterative solvers are equipped with robust and efficient preconditioners. This is the focus of the next chapters where the performance of newly proposed and popular preconditioners is analysed in depth.

# Chapter 6

# Solution Strategies for Stochastic Galerkin Methods - Linear Stochastic Case

## 6.1 Introduction

The scope of this chapter is to review the state-of-the-art solvers for the discrete linear systems obtained from the Stochastic Galerkin methods presented in Chapter 4. Similar studies have been carried out by other researchers, see for example Rossell et al. (2008), Furnival (2008), Powell & Elman (2009), Ernst et al. (2009), Elman et al. (2010) and Rossell & Vandewalle (2010).

We study the efficiency of the conjugate gradient ($CG$) and minimal residual ($MINRES$) solvers when equipped with preconditioners proposed in §4.5.3 and §4.6.2. Additionally for the stochastic primal formulation (second order problem) we also look at the performance of Gauss-Seidel solvers.

As for the deterministic case, we emphasize the conditions for which $h$ and $\mathcal{C}$-

optimality are achieved. Note that the conductivity coefficient depends on the statistical parameters $\mu$ and $\sigma$ and the number of terms in the Karhunen-Loéve expansion, $d$, (see §4.3). Hence, ideally we seek a solver which is optimal with respect to all of these parameters. All the test cases reported in this chapter and Chapter 7 are based on finite element discretisations with regular connectivity, i.e. any node of the finite element mesh has the same number of neighboring nodes. Experiments aimed at assessing solver's performance on finite element meshes with irregular connectivity are not reported in this thesis.

Additionally, the size of the stochastic space (and hence the size of the discrete problem) depends on the highest order of the polynomial basis $p$. Hence a solver which is also $p$-optimal possesses a very favourable property.

The algorithms used for the numerical experiments follow the implementation initially proposed by Ghanem & Kruger (1996) whereby the linear system is never fully assembled. Only the non-zero entries of the polynomial chaos coefficients (appropriately indexed) and $d + 1$ matrices (associated with the discretisation of the spatial random field) are stored. Hence all the non-zero blocks of $A$ are computed again at every iteration. Certainly the non-zero blocks of $A$ could be stored but this would cause further memory and computational limitations on the implementation of SG methods.

The first section concerns the solution of SFEM discrete linear systems. Overall twelve different methods are analysed some of which differ only in the solver used to invert the diagonal blocks of the coefficient matrix. In particular, we use: an incomplete Cholesky factorisation of $K_0$, using the MATLAB *cholinc* function with $droptol = 10^{-4}$ (initially proposed by Pellissetti & Ghanem (2000)); a black-box solver based on *AMG* (initially proposed by Powell & Elman (2009)) and a sparse direct solver, namely *UMFPACK*.

The algebraic multigrid uses the MATLAB implementation of the *HSL_MI20* library (Boyle et al. 2007, 2009). A symmetric Gauss-Seidel algorithm is used as smoother to remove the high frequency components of the error vector. An alternative choice is damped Jacobi. However, no simulations are carried out using this smoother.

The second section proposes suitable preconditioners for the solution of the stochastic version of the mixed problem. We only analyse the efficiency of a mean-based preconditioner which uses either *UMFPACK* or *AMG* to invert the Schur complement. Other preconditioners such as the Kronecker product preconditioner and the augmented-type preconditioners, have been recently proposed (see Powell & Ullmann (2010)). However, their implementation is not trivial and they require further investigation.

The simulations have all been carried out in serial within the MATLAB environment installed in the SRIF-3 Cluster machine (Merlin) at Cardiff University.

## 6.2 SFEM solvers

### 6.2.1 Block-diagonal preconditioner

**Test problem** 1 **- variable** $h$

The boundary conditions and source term for this test problem are described in §5.2.1.

Table 6.1 reports the size of the stochastic space $P$ and the total number of unknowns associated with each value of the discretisation parameter $h$.

Table 6.2 reports iteration counts and times for CG equipped with the incomplete Cholesky (*cholinc*) and algebraic multigrid (*AMG*) versions of the block-diagonal preconditioner ($\mathcal{P}_{bdiag}$). The set-up times for the problem and the preconditioners

Table 6.1: Dimensions of $P$ and total number of unknowns - Primal Formulation

| | | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | $P$ | 15 | 35 | 70 |
| | $h = \frac{1}{32}$ | $16,335$ | $38,115$ | $76,230$ |
| | $h = \frac{1}{64}$ | $63,375$ | $147,875$ | $295,750$ |
| | $h = \frac{1}{128}$ | $249,615$ | $582,435$ | $1,164,870$ |
| $d = 6$ | | | | |
| | $P$ | 28 | 84 | 210 |
| | $h = \frac{1}{32}$ | $30,492$ | $91,476$ | $228,690$ |
| | $h = \frac{1}{64}$ | $118,300$ | $354,900$ | $887,250$ |
| | $h = \frac{1}{128}$ | $465,948$ | $1,397,844$ | $3,494,610$ |

are reported in Appendix A (Table A.1). The set-up for the preconditioners, i.e. the construction of the coarse grids and the computation of the factorisation of $K_0$, is performed only once.

Results for the *cholinc* and *AMG* versions of the mean preconditioner ($\mathcal{P}_{mean}$) are included in Appendix B (Table B.1).

Table 6.2: CG iterations and solution timings for $\mathcal{P}_{bdiag}$ - Test Problem 1

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *cholinc* | $\frac{1}{32}$ | 11 | 0.58 | 17 | 0.53 | 31 | 2.02 |
| | $\frac{1}{64}$ | 13 | 0.75 | 22 | 3.12 | 38 | 10.77 |
| | $\frac{1}{128}$ | 19 | 10.22 | 29 | 38.21 | 55 | 151.07 |
| *AMG* | $\frac{1}{32}$ | 9 | 0.61 | 10 | 0.83 | 11 | 1.8 |
| | $\frac{1}{64}$ | 9 | 0.68 | 10 | 1.79 | 11 | 4.03 |
| | $\frac{1}{128}$ | 9 | 3.11 | 10 | 8.43 | 11 | 19.18 |
| $d = 6$ | | | | | | | |
| *cholinc* | $\frac{1}{32}$ | 11 | 0.27 | 18 | 1.41 | 31 | 6.75 |
| | $\frac{1}{64}$ | 14 | 1.49 | 22 | 7.75 | 39 | 36.91 |
| | $\frac{1}{128}$ | 20 | 20.56 | 29 | 92.34 | 55 | 457.25 |
| *AMG* | $\frac{1}{32}$ | 9 | 0.58 | 10 | 1.99 | 11 | 5.62 |
| | $\frac{1}{64}$ | 9 | 1.26 | 10 | 4.47 | 11 | 12.89 |
| | $\frac{1}{128}$ | 9 | 5.95 | 10 | 21.16 | 11 | 60.88 |

The results from Table 7.2 can be summarised as follows:

1. The block-diagonal preconditioner is very efficient. However, this problem represents a special case in which the variance of the coefficient $\mathcal{C}$ is small;
2. The *AMG* version of the block-diagonal preconditioner is more efficient than the *cholinc* version, especially for fine discretisations;
3. The *AMG* version is $h$-optimal and $d$-optimal. Only small variations in $N_{it}$ are observed for increasing polynomial order;
4. The *cholinc* version is neither $h$-optimal nor $p$-optimal. Only small variations in $N_{it}$ are observed for increasing $d$.

The *AMG* version of the mean preconditioner (see Table B.1) is significantly less efficient than the *AMG* version of the block-diagonal preconditioner. Interestingly the same is not observed for the *cholinc* versions.

### Test problem 2 - variable $\sigma$

The specification for test problem 2 is described in §6.2.2.

For this test problem the discretization parameter is fixed, $h = \frac{1}{32}$. The dimension of the stochastic space and the total number of unknowns are reported in Table 6.1.

Table 6.3 reports CG iteration count $N_{it}$ and timings $t_{CPU}$ for varying $d$, $p$ and $\sigma$. The problem and preconditioner (both *AMG* and *cholinc* versions) set-up times are listed in Appendix A.2.

The same simulations were performed using the mean preconditioner and the results are listed in Appendix B (Table B.2).

The results in Table 6.3 can be summarised as follows:

1. The preconditioner $\mathcal{P}_{bdiag}$ is not robust with respect to the standard deviation of the spatial random field $\sigma$. Its performance deteriorates significantly with

Table 6.3: CG iterations and solution timings for $\mathcal{P}_{bdiag}$ - Test Problem 2

| | | $p=2$ | | $p=3$ | | $p=4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d=4$ | | | | | | | |
| *cholinc* | 0.3 | 17 | 0.66 | 29 | 0.94 | 46 | 3.16 |
| | 0.5 | 19 | 0.25 | 33 | 1.08 | 54 | 3.7 |
| | 0.7 | 24 | 0.32 | 43 | 1.4 | 73 | 5 |
| *AMG* | 0.3 | 10 | 0.47 | 10 | 0.91 | 11 | 2.01 |
| | 0.5 | 12 | 0.47 | 14 | 1.26 | 16 | 2.92 |
| | 0.7 | 16 | 0.61 | 20 | 1.79 | 25 | 4.59 |
| $d=6$ | | | | | | | |
| *cholinc* | 0.3 | 17 | 0.43 | 29 | 2.42 | 47 | 10.68 |
| | 0.5 | 19 | 0.48 | 34 | 2.82 | 57 | 13 |
| | 0.7 | 24 | 0.6 | 44 | 3.65 | 92 | 21.19 |
| *AMG* | 0.3 | 10 | 0.72 | 11 | 2.41 | 12 | 6.82 |
| | 0.5 | 12 | 0.86 | 15 | 3.29 | 17 | 9.66 |
| | 0.7 | 16 | 1.14 | 23 | 5.04 | 33 | 18.74 |

increasing $\sigma$;

2. Both versions of the block-diagonal preconditioner are not $h$, $d$ and $p$-optimal.

Similar observations are obtained from the data associated with the mean preconditioner (see Table B.2).

For large standard deviations of $\mathcal{C}$, the matrices $K_k$ become increasingly more important as they contain information on the fluctuations of the spatial random field. That information is not included in the $\mathcal{P}_{bdiag}$ and $\mathcal{P}_{mean}$ preconditioners. Hence, their performance worsens for increasing $\sigma$.

### Test problem 3 - discontinuous-istropic conductivity field

In this test problem the domain $D = [0,1] \times [0,1]$ is partitioned into four subdomains namely: $D_1 = [0.0, 0.5] \times [0.0, 0.5]$, $D_2 = [0.5, 1.0] \times [0.0, 0.5]$, $D_3 = [0.5, 1.0] \times [0.5, 1.0]$ and $D_4 = [0.0, 0.5] \times [0.5, 1.0]$. A Karhunen-Loève expansion with exponential covariance and correlation lengths $l_x = l_y = 0.5$ is performed for each sub-domain.

Dirichlet boundary conditions are imposed on the left and right edge of the square domain such that $\Gamma_D = \{0,1\} \times [0,1]$. Homogeneous Neumann boundary conditions are imposed on the upper and lower edges of the domain.

The conductivity coefficient $\mathcal{C}$ is a spatially discontinuous uniform random field. Thus, independent and uniformly distributed random variables, defined in the interval $(-1,1)$, are used in this test problem. Hence the basis functions for the stochastic space are $d$-variate Legendre polynomials of order less than or equal to $p$.

Four cases are analysed, three of which have constant coefficient of variation $\delta$ and one with spatially variable $\delta$. The Gaussian distributions have the following statistical parameters:

$$1^{st}\text{ CASE}\begin{cases}\mu_{D_1} = 0.1, \sigma_{D_1} = 0.03, \mu_{D_2} = 100, \sigma_{D_2} = 30,\\ \mu_{D_3} = 1000, \sigma_{D_3} = 300, \mu_{D_4} = 1, \sigma_{D_4} = 0.3;\end{cases}$$

$$2^{nd}\text{ CASE}\begin{cases}\mu_{D_1} = 0.1, \sigma_{D_1} = 0.05, \mu_{D_2} = 100, \sigma_{D_2} = 50,\\ \mu_{D_3} = 1000, \sigma_{D_3} = 500, \mu_{D_4} = 1, \sigma_{D_4} = 0.5;\end{cases}$$

$$3^{rd}\text{ CASE}\begin{cases}\mu_{D_1} = 0.1, \sigma_{D_1} = 0.07, \mu_{D_2} = 100, \sigma_{D_2} = 70,\\ \mu_{D_3} = 1000, \sigma_{D_3} = 700, \mu_{D_4} = 1, \sigma_{D_4} = 0.7;\end{cases}$$

$$4^{th}\text{ CASE}\begin{cases}\mu_{D_1} = 0.1, \sigma_{D_1} = 0.07, \mu_{D_2} = 100, \sigma_{D_2} = 50,\\ \mu_{D_3} = 1000, \sigma_{D_3} = 600, \mu_{D_4} = 1, \sigma_{D_4} = 0.7;\end{cases}$$

The discretisation parameter is fixed, $h = \frac{1}{32}$, and the size of the problem is given in Table 6.1. Iteration counts and timings for CG preconditioned with the *AMG* and *cholinc* versions of $\mathcal{P}_{bdiag}$ and $\mathcal{P}_{mean}$ are given in Table 6.4 and Appendix B (Table B.3), respectively. The set-up time for the problem (building stiffness matrices and polynomial coefficients) and the preconditioners is reported in Appendix A (Table

A.3).

Table 6.4: CG iterations and solution timings for $\mathcal{P}_{bdiag}$ - Test Problem 3

| | $\delta = \frac{\sigma}{\mu}$ | $p=2$ | | $p=3$ | | $p=4$ | |
|---|---|---|---|---|---|---|---|
| | | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d=4$ | | | | | | | |
| | 0.3 | 20 | 0.28 | 33 | 1.02 | 53 | 3.45 |
| *cholinc* | 0.5 | 23 | 0.29 | 39 | 1.2 | 61 | 3.97 |
| | 0.7 | 25 | 0.31 | 44 | 1.39 | 74 | 4.81 |
| | 0.7,0.5,0.6,0.7 | 24 | 0.3 | 42 | 1.29 | 69 | 4.52 |
| | 0.3 | 8 | 0.68 | 9 | 0.66 | 10 | 1.52 |
| *AMG* | 0.5 | 10 | 0.31 | 12 | 0.88 | 13 | 1.94 |
| | 0.7 | 13 | 0.4 | 17 | 1.23 | 20 | 2.99 |
| | 0.7,0.5,0.6,0.7 | 12 | 0.37 | 15 | 1.09 | 17 | 2.52 |
| $d=6$ | | | | | | | |
| | 0.3 | 19 | 0.45 | 33 | 2.58 | 55 | 12.01 |
| *cholinc* | 0.5 | 21 | 0.5 | 37 | 2.91 | 62 | 13.42 |
| | 0.7 | 25 | 0.59 | 46 | 3.63 | 83 | 18.01 |
| | 0.7,0.5,0.6,0.7 | 25 | 0.59 | 43 | 3.37 | 75 | 16.25 |
| | 0.3 | 8 | 0.46 | 9 | 1.61 | 10 | 4.67 |
| *AMG* | 0.5 | 10 | 0.57 | 13 | 2.32 | 15 | 6.97 |
| | 0.7 | 13 | 0.74 | 19 | 3.42 | 25 | 11.62 |
| | 0.7,0.5,0.6,0.7 | 12 | 0.68 | 16 | 2.85 | 21 | 9.76 |

Remarks on the data presented in Table 6.4 are very similar to those summarised for test problem 2. However, it should also be noted that the data for test problem 3 shows that the preconditioners are robust with respect to discontinuities in the mean value of $\mathcal{C}$. In fact, if we compare the number of iterations for the case of $\delta = 0.3$ (discontinuous field) and the case of $\sigma = 0.3$ in test problem 2 (continuous field), it is easily understood that discontinuities have little or no negative impact on the performance of the solver. The same conclusions are inferred for all other cases.

### 6.2.2 Block Symmetric Gauss-Seidel Preconditioner

The block symmetric Gauss-Seidel preconditioner ($\mathcal{P}_{bSGS}$) is proposed to overcome some of the limitations of the popular mean and block-diagonal preconditioners. Each

Gauss-Seidel internal iteration includes a forward and backward sweep to guarantee the symmetry of the preconditioner for CG. The algorithm used in the experiments is described in §4.5.3. A fixed number of iterations $maxitb$ is used as stopping criteria. For the results listed in this section we use $maxitb = 2$.

Note that only experiments based on the symmetric version of the Gauss-Seidel algorithm are presented in this chapter. In fact, the theory of the Conjugate Gradient method (Saad 2003) requires the preconditioner to be symmetric and positive definite. The implementation of a non-symmetric Gauss-Seidel preconditioner for CG is straightforward however it was decided to not carry out experiments using such solver as this would be inconsistent with theoretical concepts.

The immediate advantage of the $bSGS$ preconditioner is the fact that all blocks of $A$ are included in the preconditioned system. Therefore, it is expected to perform well for the case in which the standard deviation of the spatial random field $\sigma$ is large.

As for $\mathcal{P}_{bdiag}$ and $\mathcal{P}_{mean}$, the preconditioner's sub-systems can be approximately inverted using either an incomplete Cholesky factorisation of $K_0$ or one V-cycle of $AMG$ code. The results of the previous section show that the $AMG$ version always outperforms the *cholinc* one. Therefore, in this section we replace the latter method with a multi-frontal sparse direct solver UMFPACK.

The settings for each of the test problems are as described in §6.2.1.

### Test problem 1 - variable $h$

Table 6.5 lists the number of iterations $N_{it}$ and the CPU time $t_{CPU}$ for CG equipped with the $UMFPACK$ and $AMG$ versions of $\mathcal{P}_{bSGS}$. The set-up time for the problem and preconditioner ($AMG$ case only) is reported in Appendix A (Table A.4). The $UMFPACK$ version of the preconditioner does not require any set-up time as the coefficient matrix is inverted exactly. The $AMG$ version instead requires the construction of the coarse grids and smoother for the multigrid approximation. However, this is performed only once.

Table 6.5: CG iterations and solution timings for $\mathcal{P}_{bSGS}$ - Test Problem 1

| | | $p=2$ | | $p=3$ | | $p=4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d=4$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 3 | 0.47 | 4 | 0.5 | 4 | 1.03 |
| | $\frac{1}{64}$ | 3 | 0.74 | 4 | 2.4 | 4 | 4.82 |
| | $\frac{1}{128}$ | 3 | 5.8 | 4 | 14.43 | 4 | 29.26 |
| *AMG* | $\frac{1}{32}$ | 6 | 0.82 | 6 | 0.88 | 6 | 1.8 |
| | $\frac{1}{64}$ | 6 | 0.83 | 6 | 2.03 | 6 | 4.22 |
| | $\frac{1}{128}$ | 6 | 3.62 | 6 | 8.87 | 6 | 18.36 |
| $d=6$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 3 | 0.3 | 4 | 1.24 | 4 | 3.22 |
| | $\frac{1}{64}$ | 3 | 1.41 | 4 | 5.9 | 4 | 15.03 |
| | $\frac{1}{128}$ | 4 | 11.42 | 4 | 35.14 | 4 | 89.92 |
| *AMG* | $\frac{1}{32}$ | 6 | 0.69 | 6 | 2.17 | 6 | 5.64 |
| | $\frac{1}{64}$ | 6 | 1.62 | 6 | 5.21 | 6 | 13.75 |
| | $\frac{1}{128}$ | 6 | 6.91 | 6 | 22.15 | 6 | 58.24 |

The results presented in Table 6.5 can be summarised as follows:

1. Both versions of the Gauss-Seidel preconditioner considerably reduce the number of CG iterations;

2. The comparison of the data with those presented in Table 6.2 show that the *AMG* version of $\mathcal{P}_{bSGS}$ only slightly improves the solution times of the block-diagonal preconditioner;

3. The UMFPACK version is more efficient than the AMG one only for very coarse meshes. This is a reflection of the fact that the latter methodology has the advantage that the computational cost grows linearly with the problem size.

### Test problem 2 - variable $\sigma$

Table 6.6 lists the iteration count and solution times for test problem 2. The problem set-up time and the *AMG* cost are reported in Appendix A (Table A.5).

Table 6.6: CG iterations and solution timings for $\mathcal{P}_{bSGS}$ - Test Problem 2

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | 0.3 | 3 | 0.18 | 3 | 0.4 | 3 | 0.82 |
| | 0.5 | 4 | 0.22 | 4 | 0.53 | 4 | 1.09 |
| | 0.7 | 5 | 0.28 | 6 | 0.79 | 8 | 2.18 |
| *AMG* | 0.3 | 6 | 0.48 | 6 | 1 | 6 | 2.05 |
| | 0.5 | 7 | 0.48 | 7 | 1.16 | 7 | 2.39 |
| | 0.7 | 8 | 0.55 | 9 | 1.5 | 10 | 3.4 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | 0.3 | 3 | 0.32 | 3 | 1 | 3 | 2.58 |
| | 0.5 | 4 | 0.42 | 4 | 1.32 | 5 | 4.27 |
| | 0.7 | 5 | 0.52 | 7 | 2.31 | 10 | 8.49 |
| *AMG* | 0.3 | 6 | 0.79 | 6 | 2.45 | 6 | 6.35 |
| | 0.5 | 7 | 0.92 | 7 | 2.86 | 8 | 8.44 |
| | 0.7 | 8 | 1.05 | 10 | 4.08 | 13 | 13.75 |

The results presented in Table 6.6 can be summarised as follows:

1. Both versions of the block symmetric Gauss-Seidel preconditioner show a significant improvement in terms of the number of CG iterations. This improvement becomes more evident for large values of $\sigma$;

2. The comparison of the *AMG* data with those of Table 6.3 (block-diagonal preconditioner) reveals that the Gauss-Seidel preconditioner is generally computationally cheaper and the improvement in performance increases with larger $\sigma$;

3. The *UMFPACK* version is more efficient than the *AMG* one. However, this is due to the fact that, in this experiment, the discretisation parameter is fixed

at $h = \frac{1}{32}$. For finer discretisations (larger problems), the multigrid version is generally more efficient (see Table 6.5) than the exact version.

### Test problem 3 - discontinuous-isotropic conductivity field

Results for this test problem are very similar to test problem 2 and therefore the observations summarised for Table 6.6 are also valid for Table 6.7. However, this test problem is primarily designed to assess the effect of a discontinuous conductivity field on the performance of the solver. As for the block-diagonal and mean preconditioners we observe that 'jumps' in the conductivity coefficient have little or no impact on the solver performance. Note that in this example the mean conductivity $\mu$ varies over four orders of magnitude in the domain. However, the number of iterations of the solver tends to be lower than for the continuous case (see Table 6.6). This is associated with the large mean values used for some of the subdomains in this test problem. In the continuous case (Test Problem 2), instead, a constant mean value ($\mu_{\mathcal{C}} = 1.0$) is used everywhere in the domain.

### Performance analysis

The experiments presented so far show that using $\mathcal{P}_{bSGS}$ significantly reduces the number of CG iterations for convergence. However, this does not necessarily result in an overall improvement in the computational time. It should be noted that the performance of CG depends on the chosen stopping criteria for the Gauss-Seidel algorithm. The results for the experiments presented in the previous sections are obtained using a fixed maximum number of iterations, $maxitb = 1$. One iteration comprises one forward and one backward sweep.

In this section we look at the performance of CG when more iterations are allowed for the block symmetric Gauss-Seidel algorithm. Consider test problem 2 with fixed

Table 6.7: CG iterations and solution timings for $\mathcal{P}_{bSGS}$ - Test Problem 3

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | 0.3 | 2 | 0.2 | 2 | 0.29 | 2 | 0.56 |
| | 0.5 | 3 | 0.19 | 3 | 0.42 | 3 | 0.83 |
| | 0.7 | 4 | 0.24 | 5 | 0.68 | 5 | 1.37 |
| | 0.7,0.5,0.6,0.7 | 4 | 0.24 | 4 | 0.55 | 5 | 1.37 |
| *AMG* | 0.3 | 6 | 0.53 | 6 | 1.14 | 6 | 2.31 |
| | 0.5 | 7 | 0.58 | 7 | 1.33 | 7 | 2.68 |
| | 0.7 | 7 | 0.57 | 8 | 1.51 | 9 | 3.44 |
| | 0.7,0.5,0.6,0.7 | 7 | 0.57 | 7 | 1.33 | 8 | 3.06 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | 0.3 | 2 | 0.23 | 2 | 0.68 | 2 | 1.75 |
| | 0.5 | 3 | 0.33 | 4 | 1.32 | 4 | 3.44 |
| | 0.7 | 4 | 0.44 | 5 | 1.65 | 7 | 5.99 |
| | 0.7,0.5,0.6,0.7 | 4 | 0.44 | 5 | 1.65 | 6 | 5.15 |
| *AMG* | 0.3 | 6 | 0.91 | 6 | 2.78 | 6 | 7.08 |
| | 0.5 | 7 | 1.06 | 7 | 3.22 | 7 | 8.26 |
| | 0.7 | 8 | 1.2 | 9 | 4.13 | 11 | 12.91 |
| | 0.7,0.5,0.6,0.7 | 7 | 1.06 | 8 | 3.69 | 10 | 11.78 |

$p = 4$ and examine the performance of the algorithm for successively larger values of $maxitb$, $maxitb = 1, 2, 3, \ldots$, until only one CG iteration is required for convergence. The CG iteration count and timings for these serial experiments with $d = 4$ and $d = 6$, are reported in Table 6.8. Note that for this analysis the UMFPACK version of the preconditioner was used.

The results reported in Table 6.8 for $d = 4$, show that the best solution times are obtained for low values of $maxitb$ (specifically $maxitb = 1$). In contrast, the results for $d = 6$ suggest that for large $\sigma$ the best computational time is achieved for large values of $maxitb$ (specifically $maxitb = 21$).

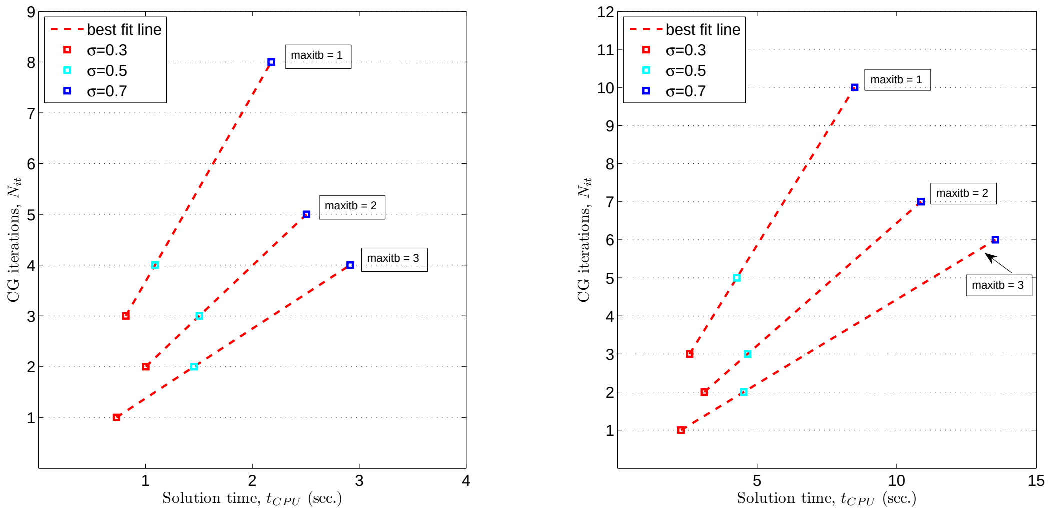
The case in which $maxitb$ is large and $t_{CPU}$ is small corresponds to the situation in which convergence is obtained in one CG iteration. It is clear that in this circumstance the bulk of the computational work is done by the preconditioner ($\mathcal{P}_{bSGS}$) and very

Table 6.8: CG iterations and solution timings (sec.) for $\mathcal{P}_{bSGS}$ for various values of *maxitb* - Test Problem 2

| | *maxitb* | $\sigma = 0.3$ | | $\sigma = 0.5$ | | $\sigma = 0.7$ | |
|---|---|---|---|---|---|---|---|
| $d = 4$ | | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | 1 | 3 | 0.82 | 4 | 1.09 | 8 | 2.18 |
| | 2 | 2 | 1.00 | 3 | 1.50 | 5 | 2.51 |
| | 3 | 1 | 0.73 | 2 | 1.45 | 4 | 2.91 |
| | 4 | 1 | 0.95 | 2 | 1.90 | 3 | 2.86 |
| | 5 | 1 | 1.18 | 2 | 2.37 | 3 | 3.55 |
| | 6 | 1 | 1.41 | 1 | 1.41 | 3 | 4.23 |
| | 8 | 1 | 1.86 | 1 | 1.86 | 2 | 3.74 |
| | 10 | 1 | 2.38 | 1 | 2.38 | 2 | 4.77 |
| | 12 | 1 | 2.80 | 1 | 2.80 | 2 | 5.62 |
| | 14 | 1 | 3.24 | 1 | 3.23 | 1 | 3.23 |
| $d = 6$ | | | | | | | |
| | 1 | 3 | 2.58 | 5 | 4.27 | 10 | 8.49 |
| | 2 | 2 | 3.11 | 3 | 4.66 | 7 | 10.87 |
| | 3 | 1 | 2.27 | 2 | 4.52 | 6 | 13.53 |
| | 4 | 1 | 2.96 | 2 | 5.93 | 5 | 14.82 |
| | 5 | 1 | 3.66 | 2 | 7.31 | 4 | 14.59 |
| | 6 | 1 | 4.36 | 2 | 8.73 | 4 | 17.44 |
| | 7 | 1 | 5.02 | 1 | 5.01 | 3 | 15.06 |
| | 10 | 1 | 7.35 | 1 | 7.36 | 3 | 21.40 |
| | 15 | 1 | 10.99 | 1 | 10.94 | 2 | 21.71 |
| | 21 | 1 | 5.98 | 1 | 5.99 | 1 | 5.99 |

little by the main solver (CG). Given that the preconditioner should serve only as a means to improve the conditioning of the system matrix, the results showing just one CG iteration should not be taken into consideration in relation to the performance analysis carried out in this section. On the other hand, this aspect reveals that an independent Gauss-Seidel (symmetric or not symmetric) solver could be a very efficient alternative to Krylov subspace iterative schemes. In §6.2.3 results obtained using Gauss-Seidel solvers are reported for all test problems considered in this chapter.

Excluding the data associated with one CG iteration, Table 6.8 show that for all values of $\sigma$ the best computational performance is achieved for $maxitb = 1$. Figures 6.1a and 6.1b show CG iterations versus CPU times for $maxitb = 1, 2, 3$ for $d = 4$ and $d = 6$, respectively. The figures indicate that there is a clear linear relationship



(a) $d = 4$

(b) $d = 6$

Figure 6.1: Performance analysis of CG preconditioned with $\mathcal{P}_{bSGS}$ for Test Problem 2

between CG (preconditioned with $\mathcal{P}_{bSGS}$), $\sigma$ and $t_{CPU}$. Out of the three best fit lines pictured the one for $maxitb = 1$ shows the best convergence rate. Given these considerations one Gauss-Seidel iteration was chosen as stopping criteria for all the numerical experiments.

### 6.2.3 Gauss Seidel Solvers

The experiments carried out for CG equipped with a block symmetric Gauss-Seidel preconditioner revealed that this methodology could also be effective when used as a stand alone solver. We perform simulations based on a symmetric ($bSGS$) and a non-symmetric ($bGS$) block Gauss-Seidel solver. The symmetric case includes a forward and a backward sweep per iteration and the non-symmetric case only a forward sweep.

As explained in §4.5.3, there are several possible re-orderings for the block structure of $A$ and a Gauss-Seidel algorithm could perform differently according to such re-orderings. Examples of reordering aimed at reducing the bandwidth of $A$ using a reverse Cuthill-McKee algorithm are given in Keese (2004). In our implementation we retain the structure as presented in Figure 4.1 and obtained by the summation of progressive ($i = 1, \ldots, d$) Kronecker terms (see 4.38). This ordering is the most natural choice as it represents the summation of decreasing Karhunen-Loéve modes (see 4.3).

As for $\mathcal{P}_{bdiag}$ and $\mathcal{P}_{bSGS}$ preconditioners, there can be various versions of the GS algorithm depending on the method used to solve the $P$ linear sub-systems of equations. In the main text we report experiments based on algebraic multigrid whilst Appendix D lists the results based on *UMFPACK*.

As for CG, the tolerance for the GS solvers is set to $10^{-8}$. In each table we list iteration count $N_{it}$ and solution times $t_{CPU}$ for both *bSGS* and *bGS*.

### Test problem 1 - variable $h$

Table 6.9 lists iteration count and solution times for test problem 1. Results from this Table are summarised as follows:

1. GS solvers are also optimal with respect to the discretisation parameter $h$;
2. Both *AMG* versions of *bSGS* and *bGS* are computationally more efficient than CG with either $\mathcal{P}_{bdiag}$ or $\mathcal{P}_{bSGS}$ preconditioners;
3. For this test problem the non-symmetric implementation of the Gauss-Seidel solver (*bGS*) is computationally more efficient than the symmetric implementation (*bSGS*).

It should be noted that the cost per iteration of bGS is approximately half that of bSGS given that bGS only performs a forward sweep. One bSGS iteration involves

Table 6.9: bSGS and bGS iterations and solution timings (*AMG* case) - Test Problem 1

| | $h$ | $p=2$ $N_{it}$ | $p=2$ $t_{CPU}$ (sec.) | $p=3$ $N_{it}$ | $p=3$ $t_{CPU}$ (sec.) | $p=4$ $N_{it}$ | $p=4$ $t_{CPU}$ (sec.) |
|---|---|---|---|---|---|---|---|
| $d=4$ | | | | | | | |
| $bSGS$ | $\frac{1}{32}$ | 4 | 0.24 | 4 | 0.53 | 4 | 1.06 |
| | $\frac{1}{64}$ | 4 | 0.48 | 4 | 1.16 | 4 | 2.34 |
| | $\frac{1}{128}$ | 5 | 2.4 | 5 | 5.69 | 5 | 11.59 |
| $bGS$ | $\frac{1}{32}$ | 6 | 0.18 | 6 | 0.4 | 6 | 0.8 |
| | $\frac{1}{64}$ | 6 | 0.37 | 6 | 0.87 | 7 | 2.02 |
| | $\frac{1}{128}$ | 6 | 1.46 | 7 | 4.04 | 7 | 8.22 |
| $d=6$ | | | | | | | |
| $bSGS$ | $\frac{1}{32}$ | 4 | 0.43 | 4 | 1.27 | 4 | 3.2 |
| | $\frac{1}{64}$ | 4 | 0.93 | 4 | 2.85 | 5 | 9.07 |
| | $\frac{1}{128}$ | 5 | 4.51 | 5 | 13.89 | 5 | 35.55 |
| $bGS$ | $\frac{1}{32}$ | 6 | 0.33 | 6 | 0.96 | 6 | 2.43 |
| | $\frac{1}{64}$ | 6 | 0.7 | 6 | 2.12 | 7 | 6.45 |
| | $\frac{1}{128}$ | 6 | 2.74 | 7 | 9.84 | 7 | 25.47 |

two sweeps. Therefore, 5 iterations of bSGS in Table 6.9, for example, corresponds to 10 sweeps. When this is compared with bGS sweeps we see that the latter is up to 30% cheaper in terms of computational time.

### Test problem 2 - variable $\sigma$

Table 6.10 lists the iteration count and solution times for test problem 2. Results for the *UMFPACK* version of the Gauss-Seidel solvers are included in Appendix D (Table D.2).

The main findings from this table can be summarised as follows:

1. Gauss-Seidel solvers are less efficient than $CG$ preconditioned with $\mathcal{P}_{bSGS}$ for all values of $\sigma$ (see Table 6.6);

2. The results of the performance analysis section seemed to indicate that a Gauss-Seidel solver would perform better as a stand alone solver than as a preconditioner for CG. However, this initial observation was not confirmed by the

Table 6.10: bSGS and bGS iterations and solution timings ($AMG$ case) - Test Problem 2

| | $\sigma$ | $p=2$ $N_{it}$ | $p=2$ $t_{CPU}$ (sec.) | $p=3$ $N_{it}$ | $p=3$ $t_{CPU}$ (sec.) | $p=4$ $N_{it}$ | $p=4$ $t_{CPU}$ (sec.) |
|---|---|---|---|---|---|---|---|
| $d=4$ | | | | | | | |
| $bSGS$ | 0.3 | 5 | 0.34 | 6 | 0.91 | 6 | 1.81 |
| | 0.5 | 8 | 0.52 | 9 | 1.36 | 10 | 3.01 |
| | 0.7 | 11 | 0.71 | 15 | 2.24 | 22 | 6.57 |
| $bGS$ | 0.3 | 7 | 0.24 | 8 | 0.61 | 8 | 1.22 |
| | 0.5 | 10 | 0.33 | 12 | 0.91 | 14 | 2.11 |
| | 0.7 | 15 | 0.49 | 22 | 1.65 | 34 | 5.13 |
| $d=6$ | | | | | | | |
| $bSGS$ | 0.3 | 6 | 0.72 | 6 | 2.17 | 6 | 5.49 |
| | 0.5 | 8 | 0.96 | 10 | 3.61 | 11 | 10.04 |
| | 0.7 | 12 | 1.43 | 19 | 6.83 | 38 | 34.65 |
| $bGS$ | 0.3 | 7 | 0.43 | 8 | 1.46 | 9 | 4.14 |
| | 0.5 | 10 | 0.61 | 13 | 2.36 | 16 | 7.34 |
| | 0.7 | 16 | 0.96 | 27 | 4.87 | 59 | 26.97 |

consequent analysis, results of which are presented in this table.

### Test problem 3 - discontinuous-isotropic conductivity field

For completeness the results of the Gauss-Seidel simulations for test problem 3 are presented in Table 6.11. These results are similar to those obtained for Test Problem 2, hence the same conclusions apply also to this problem.

Additionally, it should be noted that in contrast to the other solvers there is little or no difference in terms of $N_{it}$ and $t_{CPU}$ between the case with variable $\delta$ and the one with constant $\delta = 0.7$. Therefore, we can deduce that for spatially variable $\sigma$ the performance of the Gauss-Seidel solver is entirely dependent on the highest value of $\sigma$ in the domain.

Table 6.11: bSGS and bGS iterations and solution timings ($AMG$ case) - Test Problem 3

| | | $p=2$ | | $p=3$ | | $p=4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta=\frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d=4$ | | | | | | | |
| $bSGS$ | 0.3 | 5 | 0.38 | 6 | 1.03 | 6 | 2.05 |
| | 0.5 | 7 | 0.52 | 8 | 1.36 | 9 | 3.06 |
| | 0.7 | 10 | 0.73 | 13 | 2.2 | 17 | 5.77 |
| | 0.7,0.5,0.6,0.7 | 10 | 0.73 | 13 | 2.2 | 17 | 5.77 |
| $bGS$ | 0.3 | 7 | 0.27 | 8 | 0.69 | 8 | 1.37 |
| | 0.5 | 10 | 0.37 | 11 | 0.94 | 13 | 2.22 |
| | 0.7 | 13 | 0.48 | 19 | 1.61 | 26 | 4.43 |
| | 0.7,0.5,0.6,0.7 | 13 | 0.48 | 18 | 1.53 | 25 | 4.24 |
| $d=6$ | | | | | | | |
| $bSGS$ | 0.3 | 6 | 0.82 | 6 | 2.45 | 6 | 6.21 |
| | 0.5 | 8 | 1.09 | 9 | 3.67 | 11 | 11.35 |
| | 0.7 | 11 | 1.49 | 16 | 6.52 | 28 | 28.83 |
| | 0.7,0.5,0.6,0.7 | 11 | 1.49 | 16 | 6.51 | 28 | 28.89 |
| $bGS$ | 0.3 | 7 | 0.48 | 8 | 1.65 | 8 | 4.16 |
| | 0.5 | 10 | 0.68 | 13 | 2.66 | 15 | 7.77 |
| | 0.7 | 15 | 1.02 | 23 | 4.7 | 43 | 22.23 |
| | 0.7,0.5,0.6,0.7 | 15 | 1.02 | 23 | 4.69 | 43 | 22.23 |

## 6.3 Comparison and Conclusions

In the previous sections a large number of methods have been tested to identify the most efficient solver for the stochastic formulation of the diffusion problem (linear case). To identify the methods which are the most efficient and robust with respect to $h$, $\sigma$ and discontinuous $\mu$, the data presented in the previous tables are summarised in Figures 6.2, 6.3 and 6.4. Only the case for $p=4$ is considered and $d=4,6$. The methods included in the figures are listed below.

1. $CG$ with $\mathcal{P}_{bdiag}$ ($AMG$)
2. $CG$ with $\mathcal{P}_{bdiag}$ ($UMFPACK$)
3. $CG$ with $\mathcal{P}_{bdiag}$ ($cholinc$)
4. $CG$ with $\mathcal{P}_{mean}$ ($AMG$)
5. $CG$ with $\mathcal{P}_{mean}$ ($UMFPACK$)
6. $CG$ with $\mathcal{P}_{mean}$ ($cholinc$)

7. *CG* with $\mathcal{P}_{bSGS}$ (*AMG*)
8. *CG* with $\mathcal{P}_{bSGS}$ (*UMFPACK*)
9. *bSGS* (*AMG*)
10. *bSGS* (*UMFPACK*)
11. *bGS* (*AMG*)
12. *bGS* (*UMFPACK*)

Note that for the *AMG* and *cholinc* cases the time required to construct the grids and smoother for the approximation and the time required for the factorisation of $K_0$ is added to the solution times. The *UMFPACK* case does not require any set-up time.

Figure 6.2 shows that a block Gauss-Seidel solver with algebraic multigrid is the most efficient method for problems on medium to fine discretisations and small $\sigma$. The *UMFPACK* version is more efficient than the *AMG* one only for coarse meshes. Although *CG* with $\mathcal{P}_{bSGS}$ performs better than the block-diagonal and mean preconditioners, these methods are always outperformed by Gauss-Seidel solvers.

Figures 6.3 and 6.4 show that a conjugate gradient solver with $\mathcal{P}_{bSGS}$ (*UMFPACK* version) is the most efficient method for problems with medium / large standard deviation and discontinuous conductivity. However, this is true only for coarse discretisations which indicates that a better performance of the *AMG* version is expected for finer meshes. The performance of all versions of the Gauss-Seidel solvers deteriorates significantly when the standard deviation is large.

Generally it appears that, although Gauss-Seidel solvers perform well for variable meshes they are not robust with respect to $\sigma$. The block diagonal preconditioner (*AMG* version), part of the family of mean-based preconditioners, performs well for variable $h$, variable $\sigma$ and discontinuous conductivity. The same can be concluded for the *AMG* version of $\mathcal{P}_{bSGS}$ (method 7).

The outcome of this analysis reveals that the *AMG* version of $\mathcal{P}_{bSGS}$ is very efficient and the most robust solver considered in this work and therefore it should generally be used for the solution of SFEM linear systems (linear case). The *AMG* version $\mathcal{P}_{bdiag}$ is a valid alternative and it possesses the important advantage of being easy to implement.

## 6.4 SMFEM solvers

### 6.4.1 Schur complement preconditioner

In this section we report the performance of MINRES equipped with the Schur complement preconditioner, $\mathcal{P}_{Schur}$, described in §4.6.2. The computation of $diag(A)^{-1}$ is inexpensive and it is computed directly using the *back-slash* MATLAB functionality. As for the deterministic case the Schur complement part of the preconditioner can be solved exactly (using e.g. *UMFPACK*) or approximated by using one V-cycle of *AMG* code.

**Test problem** 1 **- variable** $h$

The settings for this test problem are described in §5.2.1. Table 6.12 reports the size of the stochastic space $P$ and the total number of unknowns for each level of discretisation. As for the primal formulation the size of the global system grows very quickly with $p$. Note that the size of the problem is significantly larger than for the primal formulation (see Table 6.1). This is obviously a consequence of the fact that with the mixed method, in addition to the element-wise potential approximation a solution for the normal fluxes at each discrete edge is also obtained. Table 6.12 reports the size of the stochastic space and the total number of unknowns associated with each value of the discretisation parameter $h$. The size of the linear systems reported in the table follow from (4.46) and (4.47).

Table 6.13 reports the MINRES iteration count and timings for test problem

Table 6.12: Dimensions of $P$ and total number of unknowns - Mixed Formulation

| | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|
| $d = 4$ | | | |
| $P$ | 15 | 35 | 70 |
| $h = \frac{1}{32}$ | $77,760$ | $181,440$ | $362,880$ |
| $h = \frac{1}{64}$ | $309,120$ | $721,280$ | $1,442,560$ |
| $h = \frac{1}{128}$ | $1,232,640$ | $2,876,160$ | $5,752,320$ |
| $d = 6$ | | | |
| $P$ | 28 | 84 | 210 |
| $h = \frac{1}{32}$ | $145,152$ | $435,456$ | $1,088,640$ |
| $h = \frac{1}{64}$ | $577,024$ | $1,731,072$ | $4,327,680$ |
| $h = \frac{1}{128}$ | $2,300,928$ | $6,902,784$ | $17,256,960$ |

1. The table reports results for experiments carried out using the exact version (*UMFPACK*) of $\mathcal{P}_{Schur}$ and the approximated version (*AMG*). The set-up times for the problem and the preconditioner are reported in Appendix E (Table E.1). The set-up time for the preconditioner, i.e. the CPU cost of constructing the coarse grids for $K_0$, is performed only once.

Table 6.13: MINRES iterations and solution timings for $\mathcal{P}_{Schur}$ - Test Problem 1

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 43 | 3.21 | 44 | 8.43 | 47 | 22.71 |
| | $\frac{1}{64}$ | 43 | 13.33 | 45 | 36.17 | 47 | 84.74 |
| | $\frac{1}{128}$ | 43 | 79.72 | 45 | 198.92 | 47 | 453.24 |
| *AMG* | $\frac{1}{32}$ | 45 | 3.19 | 48 | 8.63 | 49 | 22.34 |
| | $\frac{1}{64}$ | 45 | 9.25 | 48 | 26.96 | 51 | 67.1 |
| | $\frac{1}{128}$ | 47 | 41.02 | 48 | 108.98 | 51 | 268.15 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 43 | 6.26 | 45 | 30.21 | 47 | 135.22 |
| | $\frac{1}{64}$ | 43 | 26.83 | 45 | 103.38 | 48 | 392.67 |
| | $\frac{1}{128}$ | 43 | 148.77 | 45 | 533.73 | 48 | $1,830.21$ |
| *AMG* | $\frac{1}{32}$ | 45 | 6.35 | 48 | 30.98 | 49 | 137.85 |
| | $\frac{1}{64}$ | 47 | 19.88 | 49 | 85.32 | 51 | 346.05 |
| | $\frac{1}{128}$ | 47 | 80.69 | 49 | 325.83 | 52 | $1,276.01$ |

The results presented in Table 6.13 can be summarised as follows:

1. The Schur complement preconditioner is optimal or almost optimal with respect to $h$ and $d$. However, there is a small increase in the number of iterations for increasing $p$;

2. Although having a slightly larger iteration count, the AMG version of the preconditioner is more efficient than the exact version. Given that the preconditioner set-up time is performed only once (see Appendix E, Table E.1), its CPU cost has little impact on the overall solution timings;

3. Also note that for coarse discretisations the CPU timings are very similar. It is only for fine discretisations that the AMG version is more efficient.

Note that for the case $h = \frac{1}{128}$, the deterministic problem has size $82,176$ d.o.f. Considering $d = 6$ and $p = 4$, the dimension of the stochastic space is $P = 210$ and the global stochastic system has size $17,256,960$ d.o.f. Despite the very large size of the problem, the solution is obtained in just 56 minutes.

**Test problem 2 - variable $\sigma$**

The settings for test problem 2 are described in §6.2.2. The size of the problem for $h = \frac{1}{32}$ is given in Table 6.12. The performance of the Schur complement preconditioner for varying $\sigma$ is reported in Table 6.14. As for the previous case the set-up time for the preconditioner is performed only once. This is reported in Table E.2 together with the set-up timings for the test problem itself.

The results reported in Table 6.14 can be summarised as follows:

1. Similarly to the primal formulation (see §5.2.1), the performance of the Schur complement preconditioner significantly deteriorates for moderate and large values of $\sigma$;

Table 6.14: MINRES iterations and solution timings for $\mathcal{P}_{Schur}$ - Test Problem 2

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 47 | 3.22 | 49 | 8.66 | 51 | 22.32 |
| *UMFPACK* | 0.5 | 58 | 3.84 | 65 | 11.64 | 70 | 30.73 |
| | 0.7 | 72 | 4.78 | 91 | 16.44 | 111 | 48.81 |
| | 0.3 | 49 | 3.74 | 53 | 9.36 | 55 | 23.61 |
| *AMG* | 0.5 | 62 | 4.04 | 69 | 12.24 | 74 | 32.76 |
| | 0.7 | 77 | 5.08 | 97 | 17.12 | 119 | 51.45 |
| $d = 6$ | | | | | | | |
| | 0.3 | 48 | 6.64 | 51 | 29.29 | 52 | 114.62 |
| *UMFPACK* | 0.5 | 59 | 8.21 | 68 | 39.22 | 76 | 168.56 |
| | 0.7 | 76 | 10.55 | 100 | 57.95 | 140 | 312.71 |
| | 0.3 | 50 | 6.78 | 53 | 30.01 | 56 | 123.4 |
| *AMG* | 0.5 | 63 | 8.57 | 71 | 40.27 | 80 | 177.14 |
| | 0.7 | 80 | 10.93 | 106 | 60.23 | 147 | 326.64 |

2. The performance of the AMG and UMFPACK versions of the Schur complement preconditioner is similar. However, this is because the discretisation used for this test problem is coarse. It is expected that the difference in CPU cost will increase for finer discretisation levels.

### Test problem 3 - discontinuous-isotropic conductivity field

The settings for test problem 3 are described in §5.2.1. As for test problem 2, the discretisation level is fixed for $h = \frac{1}{32}$. The solver performance for varying $\delta$ is reported in Table 6.15. The problem and preconditioner set-up times are reported in Table E.3.

As it has already been observed for the stochastic primal formulation, the performance of the solver and preconditioners are not affected by spatial discontinuities in the conductivity field. In fact, the timings reported in Table 6.15 are comparable to those reported for the continuous test problem in Table 6.14.

Table 6.15: MINRES iterations and solution timings for $\mathcal{P}_{Schur}$ - Test Problem 3

| | | $p=2$ | | $p=3$ | | $p=4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d=4$ | | | | | | | |
| *UMFPACK* | 0.3 | 45 | 3.04 | 48 | 8.59 | 51 | 22.25 |
| | 0.5 | 56 | 3.74 | 63 | 11.25 | 69 | 30.2 |
| | 0.7 | 70 | 4.67 | 87 | 15.45 | 107 | 46.95 |
| | 0.7,0.5,0.6,0.7 | 67 | 4.47 | 83 | 14.74 | 100 | 43.88 |
| *AMG* | 0.3 | 49 | 3.78 | 52 | 9.4 | 53 | 23.19 |
| | 0.5 | 60 | 4.04 | 67 | 12.13 | 74 | 32.42 |
| | 0.7 | 76 | 5.1 | 94 | 17.03 | 115 | 50.59 |
| | 0.7,0.5,0.6,0.7 | 71 | 4.76 | 88 | 15.94 | 106 | 46.59 |
| $d=6$ | | | | | | | |
| *UMFPACK* | 0.3 | 46 | 6.33 | 49 | 27.7 | 52 | 114.94 |
| | 0.5 | 58 | 7.99 | 65 | 37.06 | 73 | 161.82 |
| | 0.7 | 72 | 9.89 | 96 | 55.19 | 131 | 292.77 |
| | 0.7,0.5,0.6,0.7 | 69 | 9.49 | 91 | 52.23 | 125 | 278.06 |
| *AMG* | 0.3 | 49 | 6.8 | 53 | 30.38 | 56 | 124.33 |
| | 0.5 | 62 | 8.62 | 70 | 40.25 | 79 | 176.2 |
| | 0.7 | 79 | 10.99 | 104 | 59.83 | 140 | 313.53 |
| | 0.7,0.5,0.6,0.7 | 74 | 10.26 | 96 | 55.23 | 130 | 293.1 |

### 6.4.2 Conclusions

The test problems reported in this chapter are all based on structure triangular meshes. The case of unstructured meshes with irregular connectivity is not presented in this work and it is matter for future work. Although structure meshes are used the of the indefinite linear system obtained by stochastic mixed finite element methods using MINRES equipped by a Schur complement preconditioner (4.57) is computationally very expensive. As expected this is more costly than solving the linear system obtained with the primal formulation since, in addition to the approximation of the potential, a solution for the normal fluxes at the finite element edges is also obtained.

Although computationally more expensive the efficient solution of SMFEM is largely dependent on the preconditioner used with the chosen iterative solver. The numerical experiments showed that the Schur complement preconditioner is $h$-optimal not only when the complement is inverted exactly but also when it is approximately inverted using one V-cycle of AMG code.

The experiments also showed that the preconditioner is not robust with respect to the conductivity coefficient. In fact, only the diagonal of the block-diagonal blocks of the velocity matrix $A$ are used in the preconditioned system. This is sufficient for conductivity coefficients possessing low standard deviations but generally inadequate for large standard deviations. In the latter case, in fact, the off-diagonal blocks of $A$ become significantly more important and these are not included in the preconditioned system.

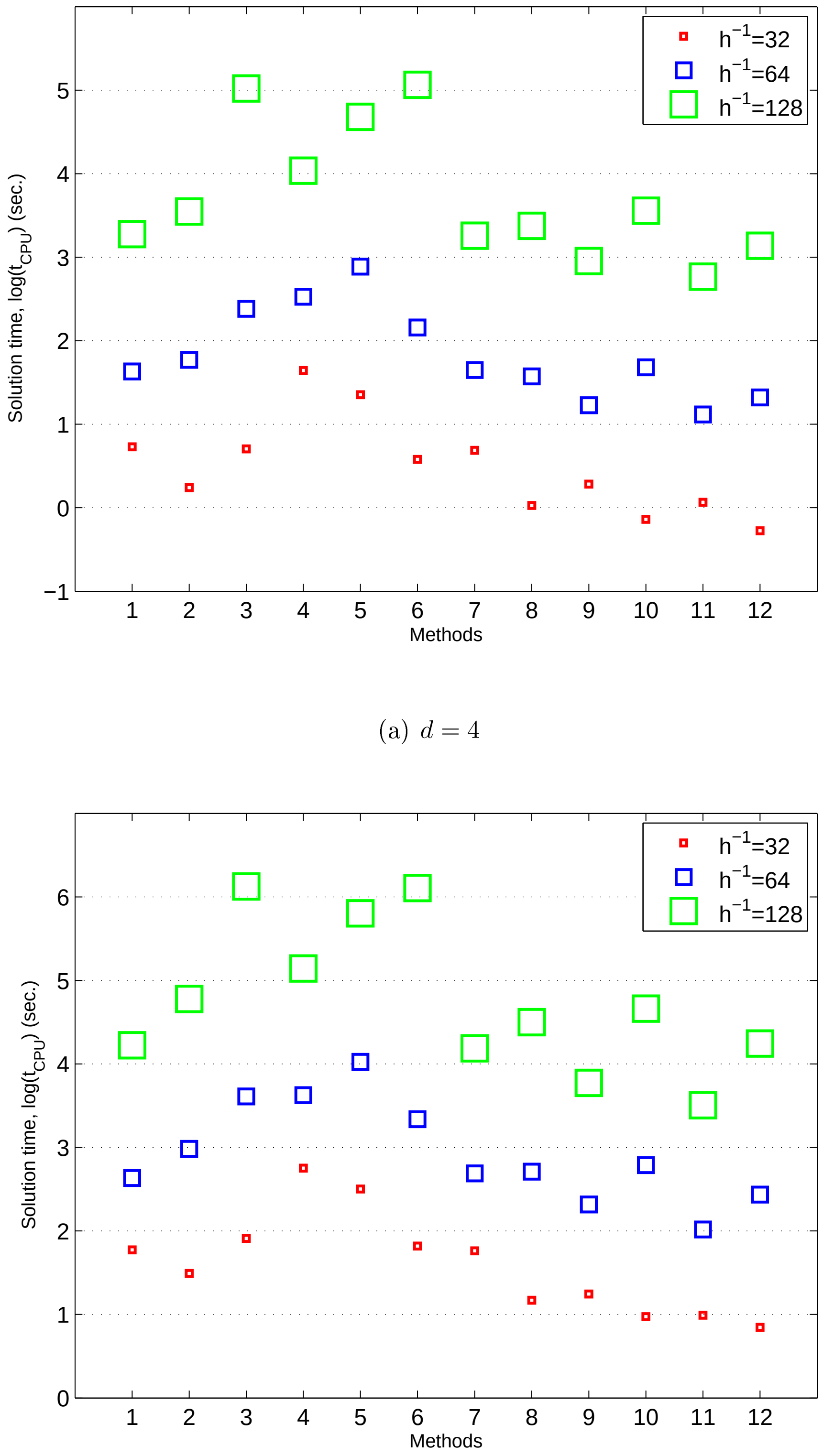
This drawback is similar to the one we faced using the mean-based preconditioner for the solution of linear systems obtained by SFEM. In those circumstances we successfully proposed a way of including the off-diagonal blocks of the coefficient matrix $A$ by means of a symmetric block Gauss-Seidel algorithm. Unfortunately, due to the structure (and specifically the presence of a zero-block) of the coefficient matrix $C = \left[\begin{smallmatrix} A & B^T \\ B & 0 \end{smallmatrix}\right]$, the same approach can not be used for the Schur complement preconditioner.

The efficient solution of discrete linear systems obtained from stochastic mixed formulations is currently a very active research area. The Kronecker product preconditioner proposed by Ullmann (2008) significantly reduces MINRES iteration counts. However, this does not necessarily corresponds to improvements in CPU performance. In fact, the author shows that the Schur complement preconditioner performs better (in terms of CPU cost) than the Kronecker preconditioner also for test problems in which the conductivity coefficient possesses large standard deviation.

It appears that SMFEM is not a very efficient method for the approximation of normal fluxes for uncertain conductivity coefficients. The question as to which method is suitable in this situation remains unanswered.

As was pointed out in Chapter 3, there exists several other (deterministic) methods that provide accurate solutions for the normal fluxes, most of which are variations of finite volume schemes. Their associated linear systems are symmetric and positive definite as should be their stochastic counterparts. Therefore, for these methods, it should be possible to use the proposed symmetric block Gauss-Seidel preconditioner and they could represent a viable alternative to SMFEM.

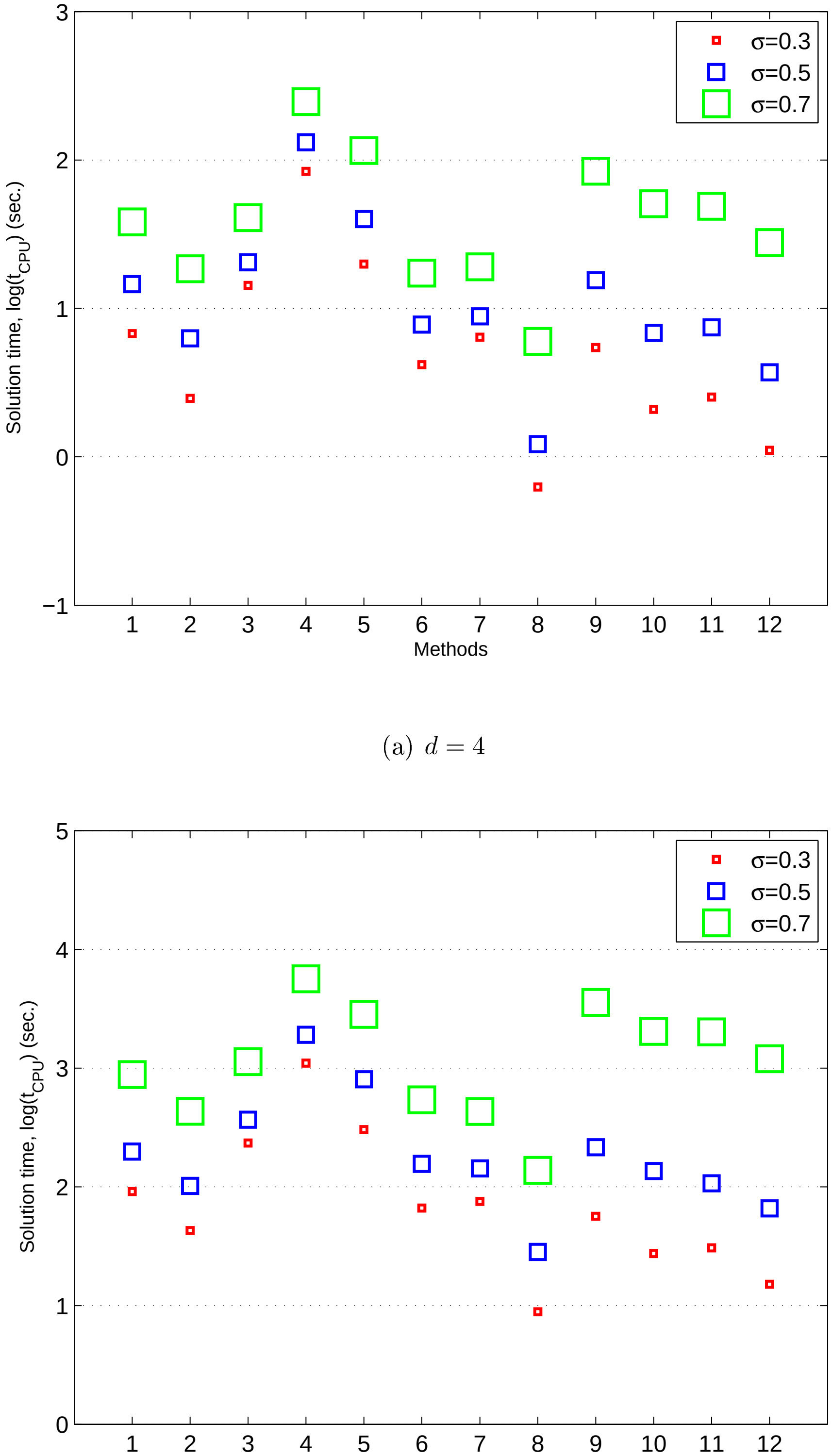
Another possibility is the stochastic implementation of the decoupling of the velocity vector from the pressure vector in (2.38) proposed by Chavent et al. (1984), Chavent & Jaffré (1986) and Scheichl (2001) for deterministic problems. The advantage of the latter approach is that the indefinite system is decoupled in a velocity system which is SPD and a triangular system for the potential approximation. Note that the implementation of such decoupling in the context of SG methods has not been reported in the literature.



(a) $d = 4$

(b) $d = 6$

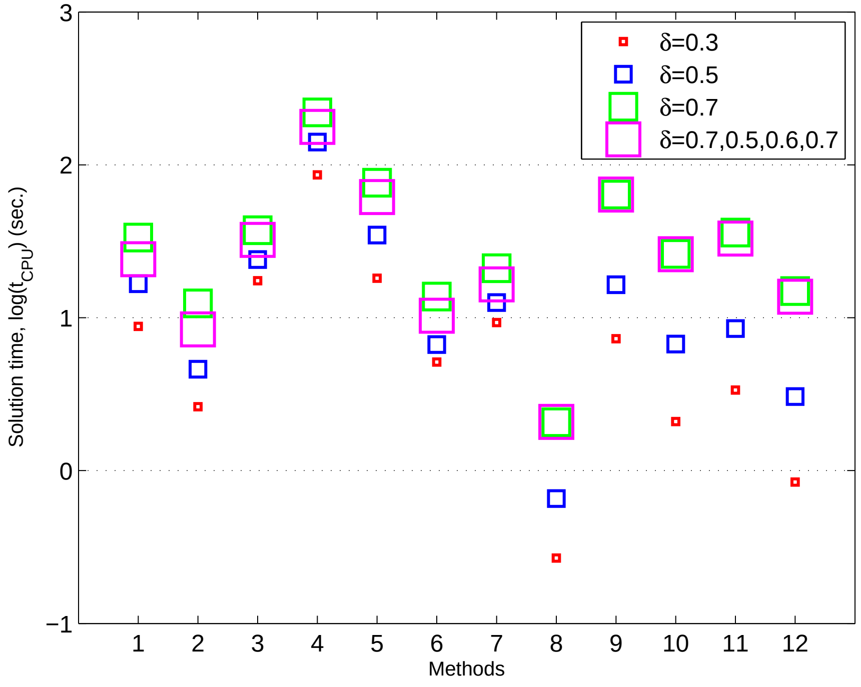
Figure 6.2: Comparison of methods for the solution of SFEM for test problem 1



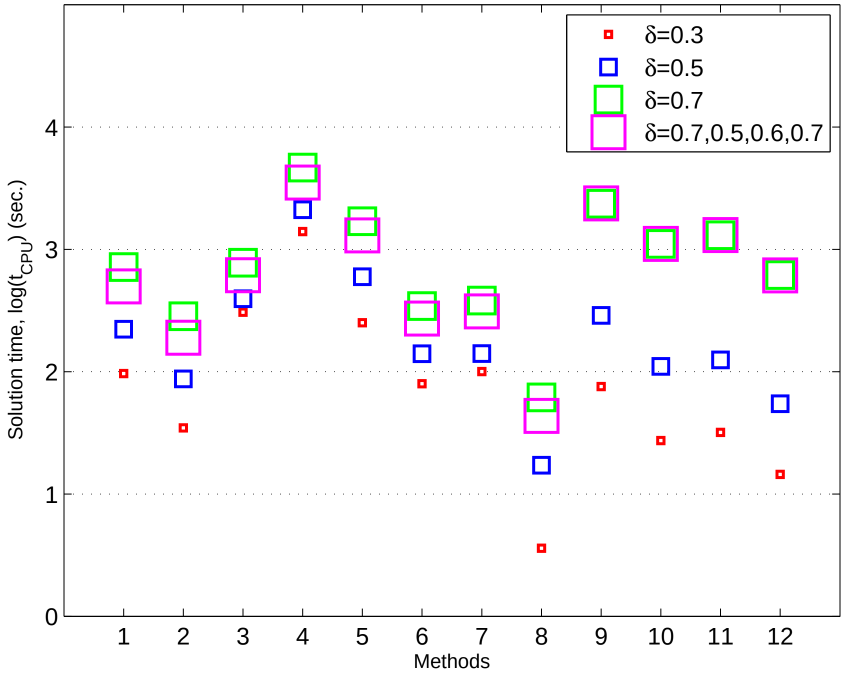
(a) $d = 4$

(b) $d = 6$

Figure 6.3: Comparison of methods for the solution of SFEM for test problem 2



(a) $d = 4$



(b) $d = 6$

Figure 6.4: Comparison of methods for the solution of SFEM for test problem 3

# Chapter 7

# Solution Strategies for Stochastic Galerkin Methods - Nonlinear Stochastic Case

## 7.1 Introduction

In the previous chapter the performance of a range of solvers was tested for problems in which the random conductivity coefficient is given in terms of a Karhunen-Loéve expansion (KLE - linear stochastic case). In this chapter the focus is on solution strategies for SG methods when the conductivity coefficient is determined by implementing a polynomial chaos expansion of a KLE.

So far we have used polynomial chaos expansions to represent unknown variables such as the potential $u$ and the normal fluxes $\mathbf{q}$. However, this expansion can also be used equally to represent input parameters (Ghanem & Spanos 2003, Sudret & Der Kiureghian 2000) such as hydraulic conductivity. More importantly, polynomial chaos expansions have been used successfully for the representation of lognormal random fields.

Lognormal random fields are very popular among physical scientists and modellers for various reasons. Firstly, there are several studies, the data of which are summarised in Gelhar (1983) and Rubin (2003), that show that parameters such as hydraulic conductivity or transmissivity are often lognormally distributed. Secondly, a lognormal distribution, although having infinite upper bound, only admits the positive part of the physical spectrum. This is obviously consistent with the physical requirement of these parameters.

The efficient solution of stochastic Galerkin problems in which the random input is a lognormal field poses important mathematical challenges. It is shown in the next section that the structure of the coefficient matrix differs significantly from the one associated with the linear case (KLE) (see Chapter 4). Specifically, for the well-posedness of $A$ to be guaranteed, the coefficient matrix is block dense, i.e there are non-zero entries for each block of $A$.

In Chapter 6 we have shown by experiments that a symmetric block Gauss-Seidel preconditioner for $CG$ represents a valid alternative to traditional mean-based preconditioners. Its advantage is that the information associated with the off-diagonal blocks of $A$ are incorporated into the preconditioned system, hence improving the conditioning of the coefficient matrix. As a result $CG$ requires few iterations to converge. The preconditioner $\mathcal{P}_{bSGS}$ is particularly efficient for those cases in which the off-diagonal blocks of $A$ hold significant information on the conductivity coefficient, i.e. problems with large values of $\sigma$.

It becomes apparent that $\mathcal{P}_{bSGS}$ should perform particularly well for the nonlinear case, given that in such circumstances $A$ is block-dense.

The author would like to express his gratitude to E. Zander for making publicly available the Stochastic Galerkin library (*sglib*) (Zander 2010) which has significantly helped the developments of the codes used in this chapter.

## 7.2 Polynomial Chaos for Lognormal Random Field

A lognormal random field is obtained by transforming a Gaussian random field. In §4.3 we have seen that a Gaussian random field can be approximated by a truncated Karhunen-Loéve expansion. Its exponentiation gives a lognormal random field

$$\mathcal{L}(\mathbf{x}, \xi(\omega)) = \exp\left(\mu(\mathbf{x}) + \sigma \sum_{i=1}^{d} \sqrt{\lambda_i}\xi_i\beta_i(\mathbf{x})\right). \tag{7.1}$$

In the context of SG methods, (7.1) is expanded by projecting the $d$ terms of the KL expansion (of the Gaussian random field) onto order $p$ polynomial chaos

$$\mathcal{L}(\mathbf{x}, \xi(\omega)) = \sum_{k=1}^{P} L_k(\mathbf{x})\chi_k(\xi), \tag{7.2}$$

where $L_k(\mathbf{x})$ are deterministic functions derived from (7.1) and for which closed forms can be obtained algebraically (see Ghanem (1999$a$,$b$), Sudret & Der Kiureghian (2000), Ghanem & Spanos (2003), Ullmann (2008)). Here $\chi_k$ are chaos polynomials in $d$ random variables (normal random variables) of degree less than or equal to $p$.

Following the discussion presented in §4.5.2, the solution vector is represented by a polynomial chaos expansion in $d$ random variables with chaos order less than or equal to $p$, as described in (4.30). Substituting (7.2) and (4.30) into the discrete variational formulation (4.22), we obtain the following Galerkin matrix, $A$

$$A = \sum_{k=1}^{N} G_k \otimes K_k, \tag{7.3}$$

where the stochastic Galerkin matrices $G_k$ are given by

$$G_k(i,j) = \langle \chi_k \chi_i \chi_j \rangle \qquad k,i,j = 1,\ldots,P, \tag{7.4}$$

and $K_k$ are deterministic matrices obtained from the discretisation of the lognormal field (7.2).

The implementation given so far considers the same maximum degree of polynomials $p$ for the polynomial chaos expansion of the solution vector $u$ and the conductivity field $\mathcal{L}(\mathbf{x},\cdot)$. In actual fact, polynomials of different orders can and should be used for the two expansions. In fact, it can be shown that for the Galerkin matrix $A$ to be positive definite (Keese 2004, Matthies & Keese 2005, Ullmann 2008) all polynomials of degree less than or equal to $2 \times p_u$ have to be included in the polynomial chaos expansion of $\mathcal{L}$, where the subscript $u$ refers to the maximum polynomial order chosen for the solution vector. Only when this condition is satisfied, a full Galerkin projection of the polynomial chaos expansion of $\mathcal{L}$ obtained. Following Ullmann (2008) the number of chaos polynomials used for the representation of $\mathcal{L}$ is

$$N = \frac{(d + 2p_u)!}{d!2p_u!}, \tag{7.5}$$

where $d$ is the number of random variables used in the Karhunen-Loéve Expansion and $p_u$ is the maximum polynomial order used for the polynomial chaos expansion of the solution vector. Note that $N$ corresponds to the number of Kronecker products in (7.3). Note that the size of the stochastic space associated with the solution vector $u$ maintains its size corresponding to $P = \frac{(d+p_u)!}{d!p_u!}$. Hence, the stochastic Galerkin matrices $G_k$ are

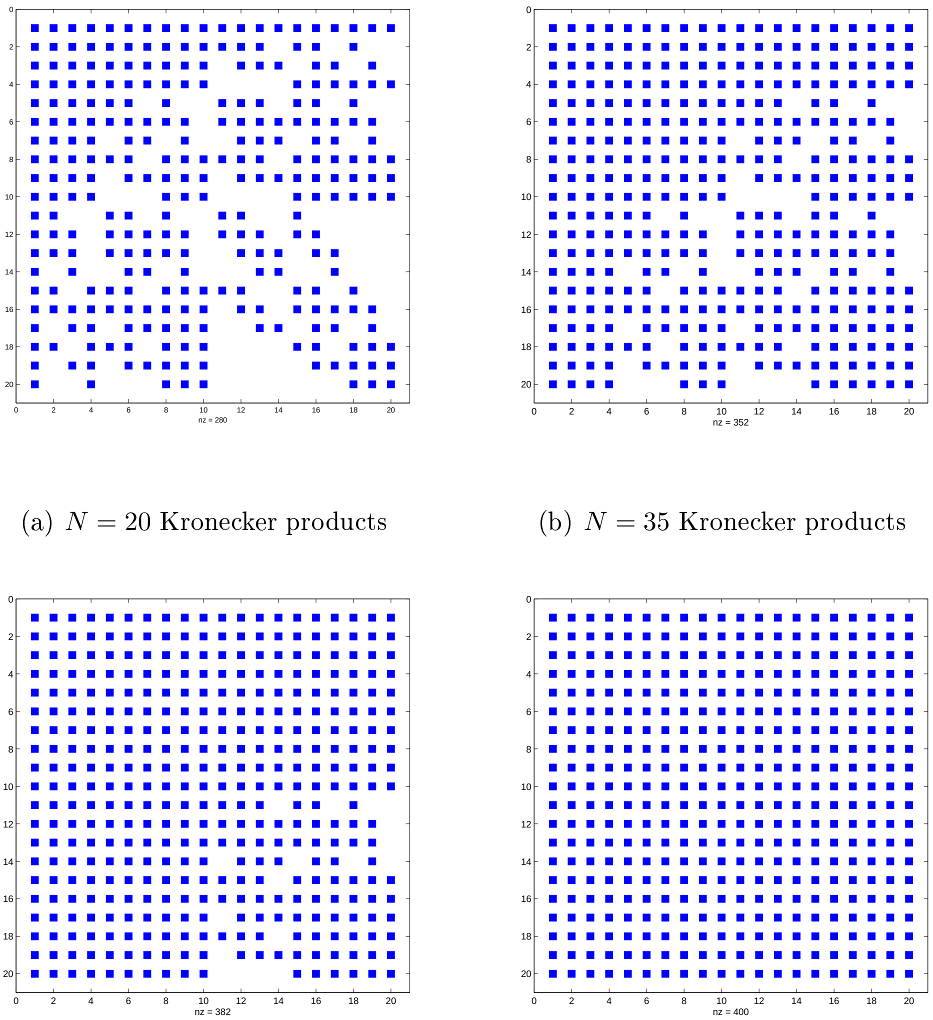
$$G_k(i,j) = \langle \chi_k \chi_i \chi_j \rangle \qquad k = 1, \ldots, N \text{ and } i,j = 1, \ldots, P. \tag{7.6}$$

It can be demonstrated (see Keese (2004) and Ullmann (2008)) that the inner product $\langle \chi_k \chi_i \chi_j \rangle$ is non-zero in only finitely many cases. In fact $\langle \chi_k \chi_i \chi_j \rangle = 0$ for all $\chi_k$ with total degree greater than $2 \times p_u$. A consequence of this observation is that given a fixed number of random variables $d$, the infinite polynomial chaos expansion of $\mathcal{L}$ automatically truncates itself as part of the SG method (see Figure 7.1(d)). Hence, since the expansion truncates naturally, no error is introduced in the representation of $\mathcal{L}$.

To make this clearer, let us consider the case in which $d = 3$ (three random variables) and $p_u = 3$ (maximum polynomial order for the solution vector). According to (4.23), the size of the stochastic space for the solution $u$ is $P = 20$. Therefore, the size of each stochastic Galerkin matrix $G_k$ in (7.3) is $20 \times 20$. Now, the number of Kronecker products $N$ can take the value 20 if the same maximum polynomial order, $p_{\mathcal{L}}$, is used for the expansion of the lognormal conductivity coefficient. Alternatively, maximum polynomial orders of 4, 5, or 6 can be used to give the number of Kronecker products corresponding to 35, 56 or 84, respectively. Although, any value of $p_{\mathcal{L}}$ can be used, it is only for $p_L = 6$ ($p_L = 2 \times p_u$), which corresponds to $N = 84$, that a full Galerkin projection of the lognormal random field is obtained. Furthermore, only in this circumstance is the global Galerkin matrix $A$ guaranteed to be positive definite (see Ullmann (2008, Remark 2.3.4)).

Figure 7.1 illustrates the block sparsity of $A$ (which corresponds to $\sum_{k=1}^{N} G_k$) for different values of $N$. Note that if polynomials of maximum order $p_{\mathcal{L}} = 6$ are used for the chaos expansion of the conductivity coefficient, then there is an entry for every block of $A$ (see Figure 7.1(d)). If polynomials of order higher than six are considered, the $G_k$ matrices corresponding to orders higher than $2 \times p_u$ would have only zero entries.

In Chapter 6 the performance of preconditioned iterative solvers was presented for SG problems in which the conductivity coefficient is described by a Karhunen-Loéve expansion. To be able to use the same preconditioners proposed in Chapter 4 and implemented in Chapter 6 it is required that $A$ is positive-definite. This is guaranteed only if order $2p$ polynomials are used for the expansion of the lognormal field (Ullmann 2008, Table 2.1).



(a) $N = 20$ Kronecker products

(b) $N = 35$ Kronecker products

(c) $N = 56$ Kronecker products

(d) $N = 84$ Kronecker products

Figure 7.1: Block sparsity of $A$

## 7.3 Comparison of Stochastic Galerkin and Monte Carlo Methods

In this section a comparison between numerical solutions obtained by SG methods and MCM when lognormal distributions are used to describe the conductivity coefficient is reported. The comparison of solutions gives us the possibility to validate the SG numerical development in a similar manner as was reported in Chapter 5 for Gaussian and uniform distributions.

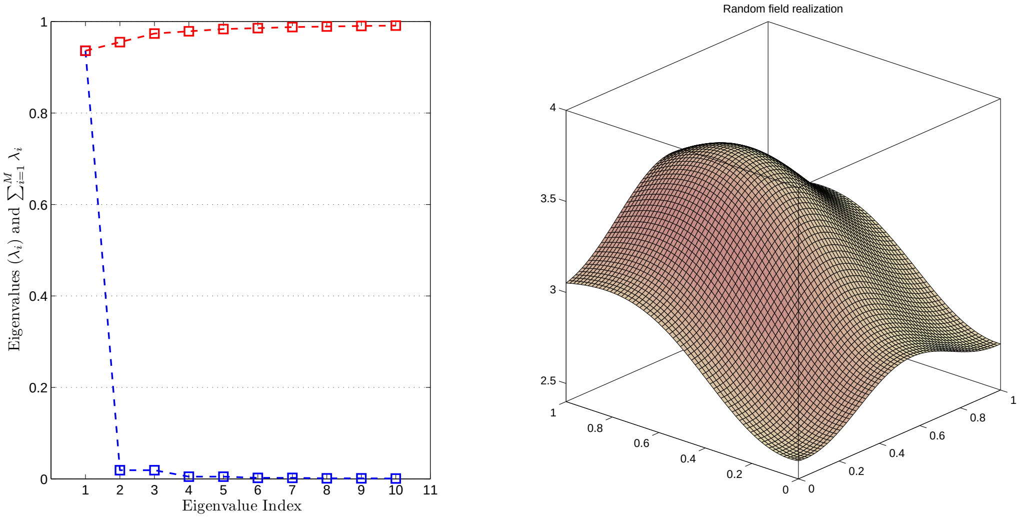
As explained in Chapter 5, the timings listed in the tables should only give the reader an indication of the CPU cost required by that specific method. The simulations have all been carried out in serial within MATLAB 7.4 on a laptop PC with $4Gb$ of RAM.

### 7.3.1 SFEM vs Monte Carlo Simulations

Consider the settings used in test problem 2 (see 6.2.2). Dirichlet boundary conditions $u = 1.0$ and $u = 0$ are imposed at the left $(x = 0)$ and right $(x = 1)$ boundaries of the model domain, respectively. Homogeneous Neumann boundary conditions $\mathcal{C}\nabla u \cdot \mathbf{n} = 0$ are imposed to the upper $(y = 1)$ and lower $(y = 0)$ edge of the model domain. Thus the dominant flow direction is from left to right.

The spatial discretisation uses a triangular mesh with $h = \frac{1}{64}$ for the approximation of $u$. This yields a total number of unknowns $N_u = 4,225$. The conductivity coefficient is a lognormal random field, $\mathcal{L} = \exp \mathcal{C}$. The underlying Gaussian random field $\mathcal{C}$ has mean, $\mu = 1$, and standard deviation, $\sigma = 0.2$. The spatial variability is modelled by an exponential correlation function with correlation lengths $l_x = l_y = 10.0$. The eigenvalues and eigenfunctions of the Karhunen-Loève expansion of $\mathcal{C}$ are available as analytical expressions (Ghanem & Spanos 2003, Powell & Elman 2009). Figure 7.2a shows the decay of the first 10 eigenvalues obtained from the KLE as well as their summation. Figure 7.2b illustrates a sample realization of the conductivity field for this test problem.

Note that the eigenvalues of the KLE of $\mathcal{C}$ decay more rapidly than for the spatial random field simulated in test problem 1, Chapter 5 (see Figure 5.1). This illustrates



(a) KLE eigenvalues for exponential covariance and $l_x = l_y = 10.0$

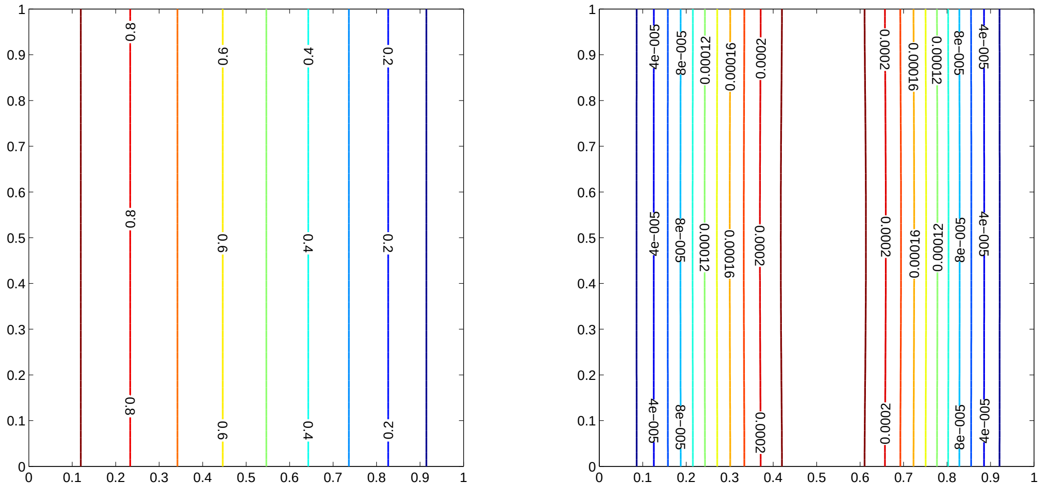
(b) Sample realization of spatial random field

Figure 7.2: KLE eigenvalues and sample realization of $\mathcal{L}(\mathbf{x}, \xi)$

that for large correlation lengths a small number of random variables (i.e. small number of terms in the KLE) are sufficient to accurately approximate the spatial random field. Conversely for small correlation lengths a limited number of random variables is generally not sufficient to fully describe the spatial random field. For this test problem we set $d = 4$ and use chaos polynomials up to order $p_u = 4$ for the solution $u$. The polynomial chaos expansion of the conductivity coefficient uses polynomials of order $2 \times p_u$ so that the positive definiteness of the coefficient matrix is guaranteed.

The mean and variance solutions for the potential obtained using SFEM with $p_u = 4$ and $d = 4$ on a $64 \times 64$ uniform grid are illustrated in Figure 7.3.

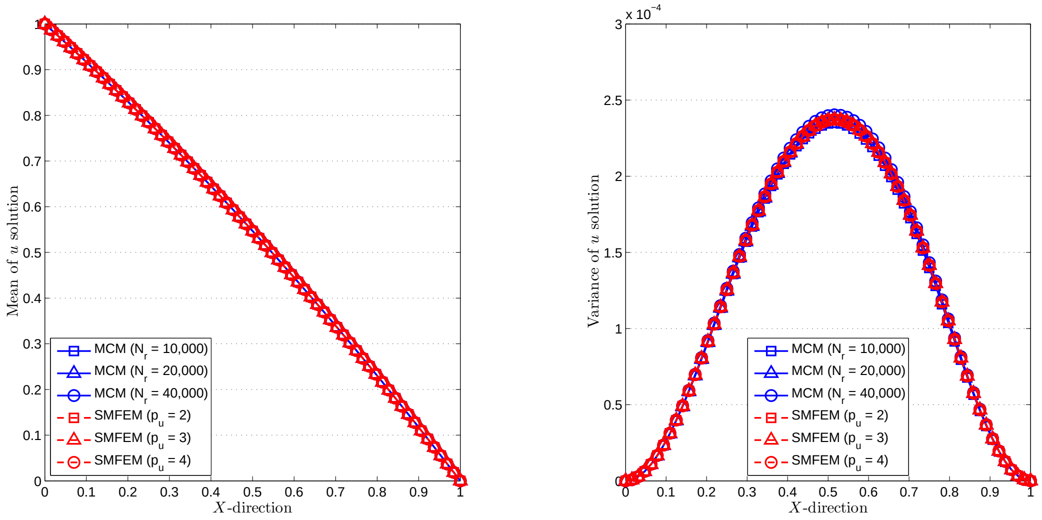
Figure 7.4 shows the mean and variance solution profiles along the horizontal centreline of the domain for several values of polynomial order $p_u$ and number of MC simulations $N_r$. As with the examples reported in Chapter 5, the solution profiles



(a) Mean $u$ solution (b) Variance $u$ solution

Figure 7.3: Mean and variance solutions for test problem with lognormal distribution (SFEM)

obtained by the two methods are very similar and tend to converge to the same values for increasing sampling size, $N_r$, and polynomial order, $p_u$.



(a) Mean $u$ solution (b) Variance $u$ solution

Figure 7.4: Comparison of solution profiles for SFEM and MCM for test problem with lognormal distribution

Table 7.1 shows the value of the mean and variance at location $(0.5, 0.5)$ for several values of $N_r$ and $p_u$. Polynomials of order two are sufficient to achieve convergence to the fourth significant digit for the mean solution. Polynomials of order three, instead, are required for the variance solution to achieve the same level of accuracy.

As for Gaussian and uniform distributions, Monte Carlo methods converge slowly. Table 7.1 shows that $20,000$ simulations are required for the sample mean to converge. Conversely the sample variance do not converge to the desired level of accuracy for the maximum sample size considered for this test problem ($N_r = 40,000$).

Table 7.1: Convergence analysis of MCM and SFEM for test problem with lognormal distribution

| | $N_r = 10,000$ | $N_r = 20,000$ | $N_r = 40,000$ |
|---|---|---|---|
| Sample Mean | $0.546\underline{8}8$ | $0.546\underline{9}6$ | $0.546\underline{9}8$ |
| Sample Variance | $0.000234\underline{6}4$ | $0.000236\underline{4}8$ | $0.000239\underline{7}9$ |
| $t_{CPU}(sec.)$ | 301 | 604 | $1,205$ |
| | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
| Mean | $0.546\underline{8}1$ | $0.546\underline{8}1$ | $0.546\underline{8}1$ |
| Variance | $0.000236\underline{5}9$ | $0.000236\underline{8}3$ | $0.000236\underline{8}3$ |
| $t_{CPU}(sec.)$ | 4.15 | 36.70 | 310.40 |

In agreement with results described in Chapter 5, the CPU times reported in Table 7.1 indicate that the SFEM method is significantly more efficient than the MCM when lognormal distributions are used.

### 7.3.2 SMFEM vs Monte Carlo Simulations

The problem settings and boundary conditions are the same as the test problem presented in the previous section. However, the first order problem is solved in this section corresponding to the system of equations described in (4.2).

The spatial discretisation uses a triangular mesh with $h = \frac{1}{64}$, thus the number of unknowns given by the mixed formulation are the sum of the number of elements, $N_e = 8,192$, and number of edges, $N_{edg} = 12,416$. The stochastic space is discretised in a similar fashion to the one described for the SFEM case, i.e. polynomial chaos up to order $p_u = 4$ are used for the potential $u$ and velocity solution $\mathbf{q}$.

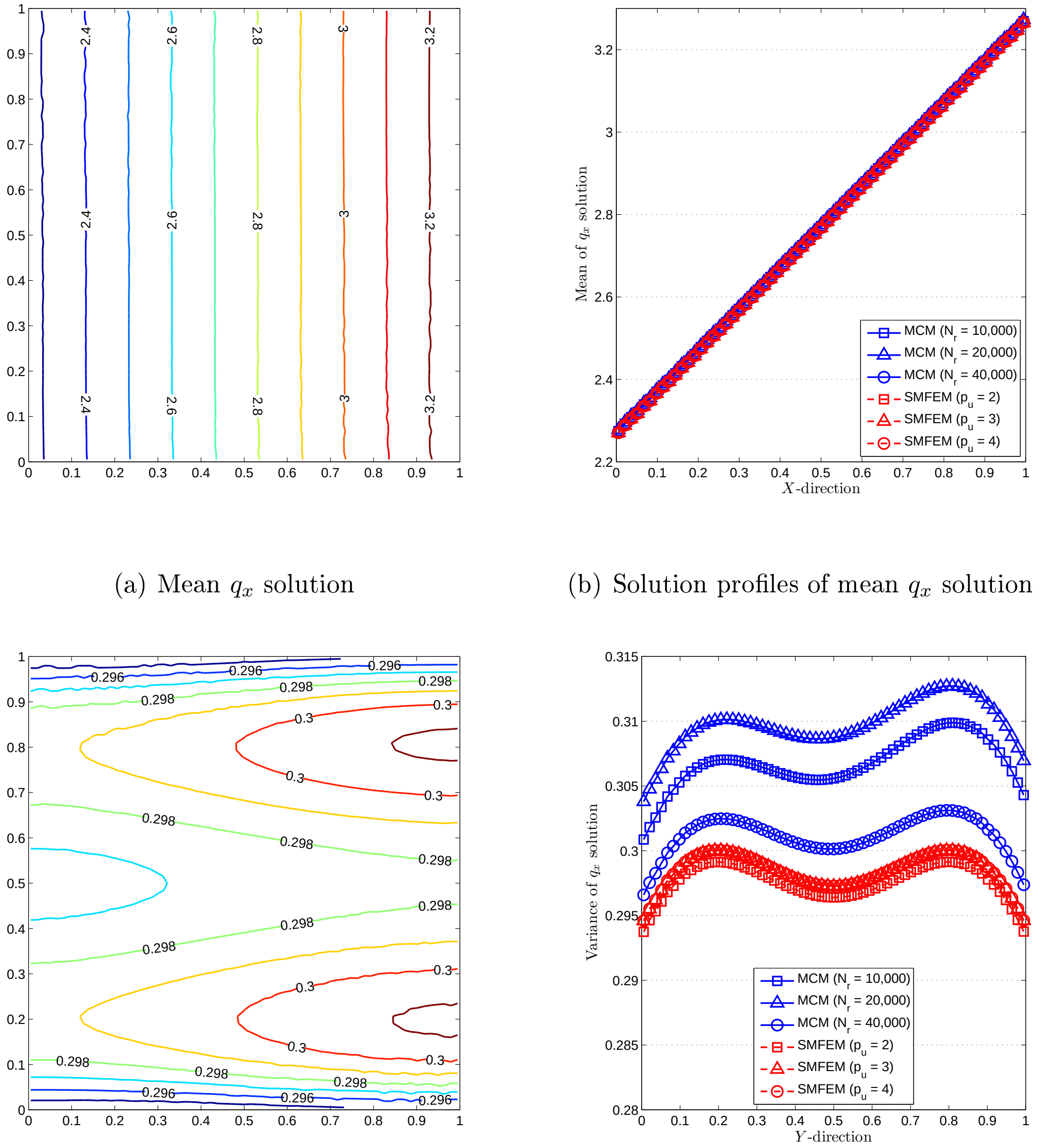
The mean and variance solution for the potential are very similar to those obtained with the second order problem and these are illustrated in Figure 7.3. The mean and variance solutions for the components of the velocity field for $d = 4$ and $p_u = 4$ on a $64 \times 64$ uniform grid are shown in Figure 7.5. The $Y$-component of the velocity field is omitted as this is close to zero (the flow is predominantly along the $X$ direction).

Figure 7.5 also includes the solution profiles for various order of polynomials $p_u$ and various Monte Carlo samples, $N_r$. For the mean velocity ($X$-component) solution the profile presented is along the direction $Y = 0.5$ and for the variance solution is along the direction $X = 0.5$. An in-depth convergence study for a sampling point having coordinate $(0.5, 0.5)$ is reported in Table 7.2.

Polynomials of order two are sufficient to achieve convergence to the fourth significant digit for the mean solution. Polynomials of order three, instead, are required for the variance solution to achieve the same level of accuracy. This is in agreement with the convergence rate of the mean and variance solution for the potential recorded for the second order problem (see Table 7.1).

It is apparent from the data presented in Table 7.2 that the Monte Carlo mean solution for the $X$ component of the velocity field does not converge for the sample size considered in this test problem. This suggests that a larger sample is required to achieve a solution with adequate accuracy. Equally, the variance solution does not converge for the maximum sample size herein considered.

Table 7.2 also includes solution timings for the MCM and SMFEM methods. Although the data show that SMFEM is more efficient than MCM this conclusion cannot be generalized. In fact the performance of preconditioned MINRES deteriorates significantly for large standard deviations of the conductivity field when lognormal



(a) Mean $q_x$ solution

(b) Solution profiles of mean $q_x$ solution

(c) Variance $q_x$ solution

(d) Solution profiles of variance $q_x$ solution

Figure 7.5: Mean and variance solutions for test problem with lognormal distribution (SMFEM)

distributions are used. Evidence of this is reported and discussed in §7.6.1.

Table 7.2: Convergence analysis of MCM and SMFEM for test problem with lognormal distribution

| | | $N_r = 10,000$ | $N_r = 20,000$ | $N_r = 40,000$ |
|---|---|---|---|---|
| $\mathbf{q}_x$ | Sample Mean | $2.771\underline{8}2$ | $2.775\underline{2}6$ | $2.770\underline{5}5$ |
| | Sample Variance | $0.305\underline{5}8$ | $0.308\underline{7}3$ | $0.300\underline{1}4$ |
| | $t_{CPU}(sec.)$ | $6,261$ | $12,524$ | $25,048$ |
| | | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
| $\mathbf{q}_x$ | Mean | $2.767\underline{8}0$ | $2.767\underline{8}2$ | $2.767\underline{8}2$ |
| | Variance | $0.296\underline{3}9$ | $0.297\underline{2}6$ | $0.297\underline{2}8$ |
| | $t_{CPU}(sec.)$ | $43.62$ | $388.21$ | $2,311.94$ |

## 7.4 SFEM solvers

As for the linear case the algorithms used for the numerical experiments are structured so that the coefficient matrix is never fully assembled. Only the non-zero entries of the polynomial chaos coefficients (appropriately indexed) and $N$ matrices (associated with the polynomial chaos discretisation of the spatial random field) are stored. In contrast to the linear case $N$ is very large, if the well-posedness of $A$ is to be guaranteed. Hence, the memory requirements of SG for the nonlinear case are significantly larger than the linear case.

In Chapter 6 it was shown that the *cholinc* version of the mean-based preconditioners, $\mathcal{P}_{bdiag}$ and $\mathcal{P}_{mean}$, is significantly less efficient than the *AMG* and *UMFPACK* versions. Hence, for the nonlinear case we do not present simulations associated with the incomplete Cholesky factorisation of $K_0$.

As for the simulations presented in Chapter 6, a symmetric Gauss-Seidel smoother is used for the AMG implementation.

The simulations have all been carried out in serial within the MATLAB environment installed in the SRIF-3 Cluster machine (Merlin) at Cardiff University. Thus the CPU timings reported in the following sections can be directly compared with those reported in Chapter 6.

### 7.4.1 Block-diagonal preconditioner

**Test problem** 1 **- variable** $h$

The settings for this test problem are as described in §5.2.1. However, the conductivity coefficient $\mathcal{L} = \exp(\mathcal{C})$ is a lognormal spatial random field. The underlying Gaussian random field has mean $\mu = 1$ and standard deviation $\sigma = 0.1$ and the same spatial model described in §5.2.1. Up to six terms of the Karhunen-Loéve expansion are used in (7.1).

Table 7.3 reports the size of the stochastic space used for the solution $u$, the total number of Kronecker products, $N$, used in the polynomial expansion of $\mathcal{L}$ and the total number of unknowns. Note that the number of Kronecker products is chosen so that the positive definiteness of $A$ is guaranteed (see (7.3) and discussion in §7.2).

Table 7.3: Dimensions of $P$, $N$ and total number of unknowns

| | | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | $P$ | 15 | 35 | 70 |
| | $N$ | 70 | 210 | 495 |
| | $h = \frac{1}{32}$ | $16,335$ | $38,115$ | $76,230$ |
| | $h = \frac{1}{64}$ | $63,375$ | $147,875$ | $295,750$ |
| | $h = \frac{1}{128}$ | $249,615$ | $582,435$ | $1,164,870$ |
| $d = 6$ | | | | |
| | $P$ | 28 | 84 | 210 |
| | $N$ | 210 | 924 | $3,003$ |
| | $h = \frac{1}{32}$ | $30,492$ | $91,476$ | $228,690$ |
| | $h = \frac{1}{64}$ | $118,300$ | $354,900$ | $887,250$ |
| | $h = \frac{1}{128}$ | $465,948$ | $1,397,844$ | $3,494,610$ |

Following the argumentation presented in Chapter 6 we report conjugate gradient performance when preconditioned by the diagonal blocks of $A$. We implement the preconditioner using either a sparse direct solver (UMFPACK) to exactly invert $K_0$ or one V-cycle of AMG code to approximately invert $K_0$. Iteration counts and solution times for various values of $d$, $p$ and $h$ are reported in Table 7.4.

Table 7.4: CG iterations and solution timings for $\mathcal{P}_{bdiag}$ - Test Problem 1

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 7 | 0.83 | 9 | 2.97 | 9 | 14.87 |
| | $\frac{1}{64}$ | 8 | 2.06 | 9 | 11.11 | 9 | 48.23 |
| | $\frac{1}{128}$ | 8 | 11.81 | 9 | 47.54 | 9 | 195.35 |
| *AMG* | $\frac{1}{32}$ | 9 | 0.63 | 10 | 3.42 | 11 | 18.26 |
| | $\frac{1}{64}$ | 9 | 1.79 | 10 | 10.99 | 11 | 56.6 |
| | $\frac{1}{128}$ | 9 | 7.17 | 10 | 42.19 | 11 | 214.86 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 7 | 1.2 | 9 | 16.05 | 9 | 123.74 |
| | $\frac{1}{64}$ | 7 | 4.77 | 9 | 53.05 | 10 | 434.61 |
| | $\frac{1}{128}$ | 7 | 21.79 | 9 | 215.67 | 10 | 1703.08 |
| *AMG* | $\frac{1}{32}$ | 9 | 1.65 | 10 | 18.13 | 11 | 152.57 |
| | $\frac{1}{64}$ | 9 | 5.16 | 10 | 55.77 | 11 | 465.32 |
| | $\frac{1}{128}$ | 9 | 19.99 | 10 | 212.22 | 11 | 1847.39 |

The set-up time for problem 1 is reported in Appendix A (Table A.4). This increases with the size of the problem. The table also includes the set-up time for the preconditioner (AMG case only), which corresponds to the computational cost of creating the coarse grids for the *AMG* approximation. This also increases with the size of the problem. However, as for the linear case this operation is implemented only once for the mean stiffness matrix, $K_0$.

Simulation results for the *UMFPACK* and *AMG* versions of the mean preconditioner, $\mathcal{P}_{mean}$, are included in Appendix B (Table B.4). The corresponding set-up times are identical to those reported for the $\mathcal{P}_{bdiag}$ case and therefore are not included in this dissertation.

The results from Table 7.4 can be summarised as follow:

1. Solution times for the nonlinear case are significantly larger than the linear case. Hence, the preconditioner set-up time (*AMG* case) becomes negligible;

2. The *UMFPACK* version of the block-diagonal preconditioner is slightly more

efficient than the $AMG$ version. For the linear case this behaviour was observed only for coarse meshes;

3. Both versions of the block-diagonal preconditioner are $d$-optimal and $h$-optimal;

4. As for the linear case the $\mathcal{P}_{mean}$ preconditioner (see B.4) is significantly less efficient than the block-diagonal one.

### Test problem 2 - variable $\sigma$

The domain size, boundary conditions and source term for this test problem are as described in §6.2.2. The conductivity coefficient $\mathcal{L}$ is a lognormal field the spatial variability of which is described in §5.2.1. The underlying Gaussian distribution has constant mean $\mu = 1$ and four different values are assigned to the standard deviation. The discretization parameter is fixed at $h = \frac{1}{32}$.

The size of the stochastic space and the total number of Kronecker products are as those reported in Table 7.3 and the the total number of unknowns corresponds to those reported in Table 7.3 for $h = \frac{1}{32}$.

Conjugate gradient, preconditioned by $\mathcal{P}_{bdiag}$, iteration count $N_{it}$ and timings $t_{CPU}$ are reported in Table 7.5. The corresponding problem and preconditioner ($AMG$ only) set-up times are given in Appendix A (Table A.5). Set-up times only depend on the size of the problem (which in this case is fixed at $h = \frac{1}{32}$), and therefore they are approximately equal for all values of $\sigma$.

The results presented in Table 7.5 can be summarised as follows:

1. The performance of the block-diagonal preconditioner deteriorates significantly for large values of standard deviation;

2. $N_{it}$ and $t_{CPU}$ show exponential growth with respect to $p$, for all values of $d$;

Table 7.5: CG iterations and solution timings for $\mathcal{P}_{bdiag}$ - Test Problem 2

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 11 | 0.7 | 14 | 4.73 | 17 | 28.66 |
| *UMFPACK* | 0.5 | 16 | 0.96 | 23 | 7.83 | 30 | 50.05 |
| | 0.7 | 23 | 1.38 | 34 | 11.52 | 48 | 80.09 |
| | 0.9 | 29 | 1.74 | 51 | 17.38 | 80 | 134.01 |
| | 0.3 | 13 | 1.32 | 16 | 5.76 | 19 | 32.56 |
| *AMG* | 0.5 | 18 | 1.23 | 25 | 8.98 | 32 | 54.67 |
| | 0.7 | 24 | 1.69 | 37 | 13.3 | 52 | 88.93 |
| | 0.9 | 31 | 2.1 | 54 | 19.33 | 83 | 142.59 |
| $d = 6$ | | | | | | | |
| | 0.3 | 11 | 1.97 | 14 | 25.48 | 17 | 239.24 |
| *UMFPACK* | 0.5 | 17 | 3 | 24 | 43.65 | 30 | 419.48 |
| | 0.7 | 23 | 4.05 | 35 | 63.48 | 50 | 697.9 |
| | 0.9 | 30 | 5.27 | 52 | 94.26 | 83 | 1158.83 |
| | 0.3 | 13 | 2.51 | 16 | 29.93 | 19 | 268.52 |
| *AMG* | 0.5 | 18 | 3.5 | 25 | 46.61 | 33 | 468.54 |
| | 0.7 | 24 | 4.6 | 38 | 70.83 | 52 | 735.08 |
| | 0.9 | 31 | 5.92 | 55 | 102.78 | 87 | 1239.99 |

3. $d$-optimality for $p_u = 4$ is lost for both versions of the preconditioner;

Simulation results for $CG$ preconditioned with $\mathcal{P}_{mean}$, are included in Appendix B (Table B.5). As for the previous case, its performance is significantly poorer than the $\mathcal{P}_{bdiag}$.

### Test problem 3 - discontinuous-isotropic conductivity field

For a description of the settings of this example refer to the corresponding test problem 3 in §5.2.1. The conductivity coefficient $\mathcal{L}$ is a spatially discontinuous lognormal random field. Four cases are presented, three of which have constant coefficient of variation $\delta$ and one with spatially variable $\delta$. The underlying Gaussian distributions (one for each of the four sub-domains) have the following parameters:

$$1^{st}\text{ CASE}\begin{cases}\mu_{D_1}=1.0,\sigma_{D_1}=0.5,\mu_{D_2}=0.1,\sigma_{D_2}=0.05,\\ \mu_{D_3}=0.01,\sigma_{D_3}=0.005,\mu_{D_4}=0.0001,\sigma_{D_4}=0.00005;\end{cases}$$

$$2^{nd}\text{ CASE}\begin{cases}\mu_{D_1}=1.0,\sigma_{D_1}=0.7,\mu_{D_2}=0.1,\sigma_{D_2}=0.07,\\ \mu_{D_3}=0.01,\sigma_{D_3}=0.007,\mu_{D_4}=0.0001,\sigma_{D_4}=0.00007;\end{cases}$$

$$3^{rd}\text{ CASE}\begin{cases}\mu_{D_1}=1.0,\sigma_{D_1}=1.0,\mu_{D_2}=0.1,\sigma_{D_2}=0.1,\\ \mu_{D_3}=0.01,\sigma_{D_3}=0.01,\mu_{D_4}=0.0001,\sigma_{D_4}=0.0001;\end{cases}$$

$$4^{th}\text{ CASE}\begin{cases}\mu_{D_1}=1.0,\sigma_{D_1}=1.0,\mu_{D_2}=0.1,\sigma_{D_2}=0.07,\\ \mu_{D_3}=0.01,\sigma_{D_3}=0.005,\mu_{D_4}=0.0001,\sigma_{D_4}=0.0001.\end{cases}$$

A Karhunen-Loéve expansion is performed for each sub-domain and the number $d$ of terms retained in the expansion is equal for each sub-domain. The case of different $d$ in each sub-domain has not been considered in this dissertation and could be a subject for further research. The same spatial model (see 5.2.1) with $l_x = l_y = 0.5$ is used for each sub-domain.

The discretisation parameter is fixed, $h = \frac{1}{32}$, and the size of the problem is given in Table 7.3. Iteration count and timings for CG preconditioned with the block-diagonal of $A$ is given in Table 7.6. The corresponding problem and preconditioner set-up times are listed in Table A.6. Also for this test problem the set-up times are approximately equal for all values of $\delta$.

Similar observations are drawn for this test problem as the ones highlighted in §7.4.1. The exponential growth of $N_{it}$ and $t_{CPU}$ with increasing $p$ is clear also for this test problem. In addition it appears that the deterioration in the performance of the preconditioner is exclusively due to the increase in the value of $\sigma$ in each sub-domain from case to case. In fact in all four cases the mean values $\mu_{D_1,D_2,D_3,D_4}$ are equal. This indicates that the preconditioned solver is robust with respect to discontinuities in the mean value of the conductivity coefficient.

Table 7.6: CG iterations and solution timings for $\mathcal{P}_{bdiag}$ - Test Problem 3

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| | 0.5 | 16 | 1.02 | 22 | 7.47 | 27 | 45.3 |
| *UMFPACK* | 0.7 | 22 | 1.33 | 32 | 10.92 | 43 | 72.27 |
| | 1.0 | 33 | 1.99 | 56 | 19.21 | 84 | 141.2 |
| | 1.0,0.7,0.5,1.0 | 33 | 1.99 | 55 | 18.73 | 84 | 140.98 |
| | 0.5 | 17 | 1.21 | 23 | 8.28 | 28 | 48.07 |
| *AMG* | 0.7 | 23 | 1.57 | 34 | 12.19 | 45 | 77.22 |
| | 1.0 | 34 | 2.32 | 57 | 20.43 | 86 | 147.55 |
| | 1.0,0.7,0.5,1.0 | 34 | 2.31 | 57 | 20.5 | 86 | 147.41 |
| $d = 6$ | | | | | | | |
| | 0.5 | 17 | 3 | 23 | 41.73 | 28 | 392.59 |
| *UMFPACK* | 0.7 | 23 | 4.08 | 34 | 61.76 | 45 | 633.19 |
| | 1.0 | 35 | 6.19 | 58 | 105.52 | 90 | 1265.67 |
| | 1.0,0.7,0.5,1.0 | 35 | 6.16 | 58 | 105.88 | 90 | 1261.79 |
| | 0.5 | 18 | 3.48 | 24 | 44.69 | 29 | 420.91 |
| *AMG* | 0.7 | 24 | 4.61 | 35 | 65.76 | 47 | 670.45 |
| | 1.0 | 36 | 6.9 | 61 | 113.85 | 93 | 1313.96 |
| | 1.0,0.7,0.5,1.0 | 36 | 6.88 | 60 | 111.57 | 93 | 1320.7 |

### 7.4.2 Block Symmetric Gauss-Seidel Preconditioner

In this section the same test problems presented in §7.4.1 are solved using a block symmetric Gauss-Seidel (bSGS) preconditioner for CG. The algorithm used in the experiments is described in §4.5.3. A fixed number of iterations *maxitb* is used as stopping criteria for $\mathcal{P}_{bSGS}$, each iteration including a forward and backward sweep. Similarly to the linear case, $maxitb = 1$ is used for the experiments reported in the following sections. The reason for this choice together with an in-depth analysis on the performance of $\mathcal{P}_{bSGS}$ for several values of *maxitb* is given in §7.4.2.

Note that only experiments based on the symmetric version of the Gauss-Seidel algorithm are presented in this chapter. In fact, the theory of the Conjugate Gradient method (Saad 2003) requires the preconditioner to be symmetric and positive definite. The implementation of a non-symmetric Gauss-Seidel preconditioner for CG is straightforward however it was decided to not carry out experiments using such solver as this would be inconsistent with theoretical concepts.

The UMFPACK implementation of the block symmetric Gauss-Seidel preconditioner is straightforward and it is identical to the one used in Chapter 6. In contrast, the AMG implementation is not straightforward and requires additional preprocessing to be implemented. In fact, differently from the linear case, the tensor products $G_k \otimes K_k, k = 1, \dots, N$, have several contributions to the blocks of the leading diagonal of the coefficient matrix, depending on the value of $d$ and $p_u$. So for example, fixing $d = 2$ and $p_u = 3$, the contributions are as follows

Table 7.7: $G_k \times K_k$ contributions to the blocks of the diagonal of $A$

| (i,j) | $G_k \times K_k$ |
|---|---|
| $(1,1)$ | |
| $(2,2)$ | $G_4(2,2) \times K_4$ |
| $(3,3)$ | $G_6 \times K_6$ |
| $(4,4)$ | $(G_4 \times K_4) + (G_{11} \times K_{11})$ |
| $(5,5)$ | $(G_4 \times K_4) + (G_6 \times K_6) + (G_{13} \times K_{13})$ |
| $(6,6)$ | $(G_6 \times K_6) + (G_{15} \times K_{15})$ |
| $(7,7)$ | $(G_4 \times K_4) + (G_{11} \times K_{11}) + (G_{22} \times K_{22})$ |
| $(8,8)$ | $(G_4 \times K_4) + (G_6 \times K_6) + (G_{11} \times K_{11}) + (G_{13} \times K_{13}) + (G_{24} \times K_{24})$ |
| $(9,9)$ | $(G_4 \times K_4) + (G_6 \times K_6) + (G_{13} \times K_{13}) + (G_{15} \times K_{15})$ |
| $(10,10)$ | $(G_6 \times K_6) + (G_{15} \times K_{15}) + (G_{26} \times K_{26}) + (G_{28} \times K_{28})$ |

Note that the $G_1$ matrix is diagonal and the product $G_1(i,j) \times K_1$ contains the mean information (this is omitted from Table 7.7). In the linear case the AMG grids are constructed only once, whereas for the nonlinear case the AMG grids have to be computed for each block entries of the diagonal of the global system. Thus the AMG pre-processing is implemented $P$ times and the grids are stored before the iterative solution process begins. Clearly the preconditioner set-up time now contributes significantly to the solver's CPU cost.

To avoid repetition we refer the reader to §7.4.1 for details on the settings of each test problem.

**Test problem** 1 **- variable** $h$

The iteration count and timings for CG preconditioned with a $\mathcal{P}_{bSGS}$ are reported in Table 7.8. The problem set-up time is listed in Appendix C (Table C.4).

Table 7.8: CG iterations and solution timings for $\mathcal{P}_{bSGS}$ - Test Problem 1

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 3 | 0.48 | 3 | 1.77 | 3 | 8.79 |
| | $\frac{1}{64}$ | 3 | 1.46 | 3 | 6.76 | 3 | 28.61 |
| | $\frac{1}{128}$ | 3 | 8.9 | 3 | 29.51 | 3 | 117.07 |
| *AMG* | $\frac{1}{32}$ | 6 | 1.11 | 6 | 3.61 | 6 | 17.37 |
| | $\frac{1}{64}$ | 6 | 2.01 | 6 | 11.12 | 6 | 52.11 |
| | $\frac{1}{128}$ | 6 | 7.22 | 6 | 40.85 | 6 | 195.45 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 3 | 0.92 | 3 | 9.6 | 3 | 74.58 |
| | $\frac{1}{64}$ | 3 | 3.84 | 3 | 31.85 | 3 | 229.6 |
| | $\frac{1}{128}$ | 3 | 17.5 | 3 | 129.57 | 3 | 901.89 |
| *AMG* | $\frac{1}{32}$ | 6 | 1.94 | 6 | 19.36 | 6 | 146.68 |
| | $\frac{1}{64}$ | 6 | 5.91 | 6 | 57.04 | 6 | 440.44 |
| | $\frac{1}{128}$ | 6 | 21.71 | 6 | 230.99 | 6 | $1,765.36$ |

The results presented in Table 7.8 can be summarised as follows:

1. The $\mathcal{P}_{bSGS}$ preconditioner is significantly more efficient than $\mathcal{P}_{bdiag}$. For example, for $h = \frac{1}{128}$ solution times are reduced by as much as 48% for $p_u = 4$, 40% for $p_u = 3$ and 13% for $p_u = 2$;

2. The number of *CG* iterations is also reduced to about a third of that required using a block-diagonal preconditioner;

3. The preconditioner is not only $h$-optimal and $d$-optimal but also $p$-optimal;

4. The UMFPACK version of the preconditioner is more efficient than the AMG version even without considering the CPU cost associated with the set-up time. If, however, we do consider the AMG set-up time, this is so large that it is actually larger than the actual solution time. Note that this applies to the non-linear case only. In fact results reported in Chapter 6, §6.2.2, for the linear case show that for fine discretisations ($\frac{1}{128}$) AMG is more efficient than UMFPACK.

### Test problem 2 - variable $\sigma$

Tables 7.9 and C.5 report CG iteration count and timings, and set-up times for test problem 2.

Table 7.9: CG iterations and solution timings for $\mathcal{P}_{bSGS}$ - Test Problem 2

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 4 | 0.87 | 4 | 2.44 | 5 | 14.95 |
| *UMFPACK* | 0.5 | 6 | 0.65 | 7 | 4.28 | 8 | 23.9 |
| | 0.7 | 8 | 0.86 | 10 | 6.11 | 12 | 35.8 |
| | 0.9 | 10 | 1.07 | 14 | 8.53 | 18 | 53.73 |
| | 0.3 | 6 | 1.11 | 7 | 4.35 | 7 | 20.82 |
| *AMG* | 0.5 | 8 | 0.94 | 9 | 5.6 | 11 | 32.63 |
| | 0.7 | 10 | 1.21 | 14 | 8.71 | 20 | 59.53 |
| | 0.9 | 14 | 1.64 | 23 | 14.27 | 46 | 136.93 |
| $d = 6$ | | | | | | | |
| | 0.3 | 4 | 1.27 | 4 | 13.11 | 5 | 123.46 |
| *UMFPACK* | 0.5 | 6 | 1.91 | 7 | 22.9 | 8 | 198.15 |
| | 0.7 | 8 | 2.55 | 10 | 32.62 | 12 | 297.21 |
| | 0.9 | 11 | 3.51 | 14 | 45.85 | 19 | 473.32 |
| | 0.3 | 6 | 2 | 7 | 22.94 | 7 | 171.93 |
| *AMG* | 0.5 | 8 | 2.68 | 10 | 32.71 | 12 | 297.72 |
| | 0.7 | 11 | 3.66 | 15 | 49.01 | 22 | 546.12 |
| | 0.9 | 15 | 5 | 25 | 81.93 | 52 | 1279.58 |

The results presented in Table 7.9 can be summarised as follows:

1. The block symmetric Gauss-Seidel preconditioner shows a significant improvement in terms of number of CG iterations. This improvement becomes more evident for large values of standard deviation ($\sigma$);

2. The comparison of the data with those of Table 7.5 (block-diagonal preconditioner) reveals that the Gauss-Seidel preconditioner is generally computationally cheaper and the improvement in performance increases with larger values of $\sigma$;

3. Noticeably the difference in performance between the exact and approximate versions of the preconditioner increases for larger $\sigma$. In fact for $\sigma = 0.9$, the AMG solution times are approximately three times larger than for UMFPACK.

### Test problem 3 - discontinuous-isotropic conductivity field

Tables 7.10 and C.6 report CG iteration count and timings, and set-up times for test problem 3 using a bSGS preconditioner.

Similarly to test problem 2 a significant improvement for both $N_{it}$ and $t_{CPU}$ is achieved. A large saving in computational cost was recorded for higher polynomial orders and large $\delta$. In fact the $t_{CPU}$ cost is reduced by 60% if compared with results obtained using the $\mathcal{P}_{bdiag}$ preconditioner. Significant time reduction is equally achieved for lower polynomial orders and coefficient of variation $\delta$. This is around 37% for $p_u = 2$ and $\delta = 0.5$, and around 53% for $p_u = 3$ and $\delta = 0.7$.

As previously observed for the $\mathcal{P}_{bdiag}$ preconditioner, it appears that the discontinuous conductivity coefficient (jumps in the mean conductivity value at the subdomains boundaries) does not worsen the preconditioner performance. It is in fact, the standard deviation which has a significant negative impact on the performance of both $\mathcal{P}_{bdiag}$ and $\mathcal{P}_{bSGS}$. Not even using the $\mathcal{P}_{bSGS}$ algorithm and therefore including the off-diagonal blocks of $A$ (which retain information on the fluctuations about the mean) can optimality of $N_{it}$ with respect to $\sigma$ be achieved.

Table 7.10: CG iterations and solution timings for $\mathcal{P}_{bSGS}$ - Test Problem 3

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| | 0.5 | 6 | 0.69 | 7 | 4.27 | 7 | 20.86 |
| *UMFPACK* | 0.7 | 8 | 0.86 | 9 | 5.54 | 11 | 33.04 |
| | 1.0 | 11 | 1.18 | 15 | 9.23 | 19 | 56.64 |
| | 1.0,0.7,0.5,1.0 | 11 | 1.19 | 15 | 9.22 | 19 | 56.73 |
| | 0.5 | 7 | 0.91 | 8 | 4.97 | 10 | 29.7 |
| *AMG* | 0.7 | 9 | 1.06 | 12 | 7.46 | 16 | 47.59 |
| | 1.0 | 14 | 1.67 | 23 | 14.43 | 48 | 143.13 |
| | 1.0,0.7,0.5,1.0 | 14 | 1.67 | 23 | 14.42 | 48 | 143.13 |
| $d = 6$ | | | | | | | |
| | 0.5 | 6 | 1.91 | 7 | 22.82 | 8 | 198.34 |
| *UMFPACK* | 0.7 | 8 | 2.55 | 9 | 29.49 | 11 | 273.95 |
| | 1.0 | 11 | 3.5 | 15 | 49.18 | 21 | 520.41 |
| | 1.0,0.7,0.5,1.0 | 11 | 3.51 | 15 | 49.08 | 21 | 523.93 |
| | 0.5 | 8 | 2.68 | 9 | 29.45 | 10 | 245.42 |
| *AMG* | 0.7 | 10 | 3.38 | 13 | 42.49 | 18 | 440.4 |
| | 1.0 | 16 | 5.41 | 27 | 89.2 | 55 | 1348.74 |
| | 1.0,0.7,0.5,1.0 | 16 | 5.39 | 27 | 88.5 | 55 | 1345.02 |

### Performance Analysis

As for the linear case, the experiments presented so far show that CG equipped with a block symmetric Gauss-Seidel preconditioner is significantly more efficient than traditional mean-based preconditioners. This conclusion depends on the stopping criteria chosen for the Gauss-Seidel algorithm. The results reported in this Chapter's tables are obtained using a maximum number of iterations, $maxitb = 1$, for the $bSGS$ algorithm. This means two sweeps per iteration, one forward and one backward to ensure the symmetry of the preconditioner for CG.

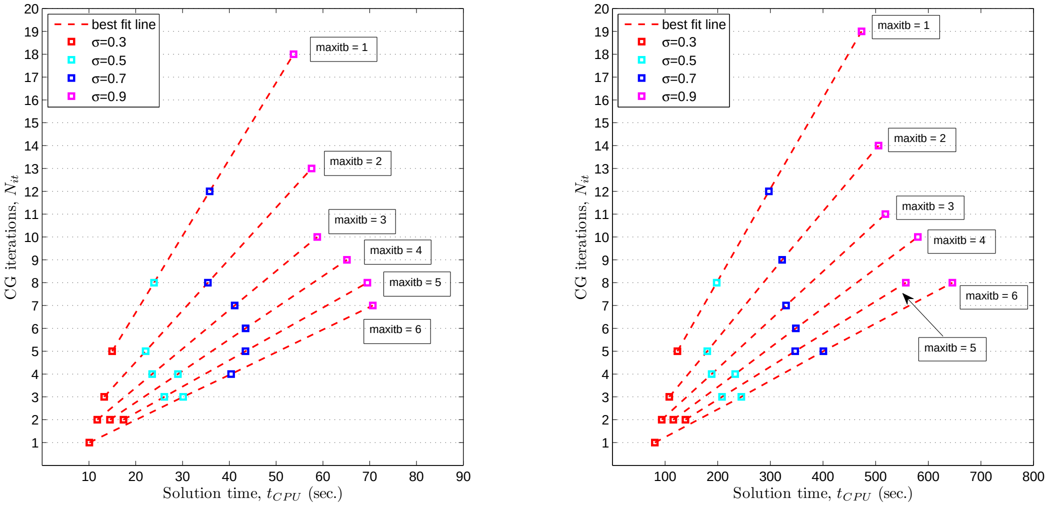
The choice of $maxitb$ can be optimized. Consider test problem 2 with fixed $p_u = 4$ and run simulations for successively larger $maxitb$, $maxitb = 1, 2, 3, \ldots$, until only one CG iteration is required for convergence. CG iteration count and timings for these experiments with $d = 4$ and $d = 6$, are reported in Table 7.11. Note that for this analysis the UMFPACK version of the preconditioner was used.

Table 7.11: CG iterations and solution timings (sec.) for $\mathcal{P}_{bSGS}$ for various values of *maxitb* - Test Problem 2

| | *maxitb* | $\sigma = 0.3$ | | $\sigma = 0.5$ | | $\sigma = 0.7$ | | $\sigma = 0.9$ | |
|---|---|---|---|---|---|---|---|---|---|
| $d = 4$ | | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | 1 | 5 | 14.95 | 8 | 23.90 | 12 | 35.79 | 18 | 53.72 |
| | 2 | 3 | 13.26 | 5 | 22.11 | 8 | 35.42 | 13 | 57.57 |
| | 3 | 2 | 11.79 | 4 | 23.50 | 7 | 41.14 | 10 | 58.79 |
| | 4 | 2 | 14.48 | 4 | 29.00 | 6 | 43.47 | 9 | 65.13 |
| | 5 | 2 | 17.39 | 3 | 26.10 | 5 | 43.45 | 8 | 69.51 |
| | 6 | 1 | 10.07 | 3 | 30.13 | 4 | 40.38 | 7 | 70.65 |
| | 8 | 1 | 12.99 | 2 | 25.87 | 4 | 51.83 | 6 | 77.51 |
| | 14 | 1 | 21.54 | 1 | 21.54 | 3 | 64.56 | 4 | 86.06 |
| | 32 | 1 | 47.26 | 1 | 47.14 | 1 | 47.17 | 3 | 141.50 |
| | 75 | 1 | 108.45 | 1 | 108.56 | 1 | 108.29 | 1 | 108.17 |
| $d = 6$ | | | | | | | | | |
| | 1 | 5 | 123.46 | 8 | 198.15 | 12 | 297.21 | 19 | 473.32 |
| | 2 | 3 | 108.02 | 5 | 180.14 | 9 | 322.50 | 14 | 505.53 |
| | 3 | 2 | 93.71 | 4 | 188.48 | 7 | 329.93 | 11 | 518.46 |
| | 4 | 2 | 115.72 | 4 | 233.33 | 6 | 348.50 | 10 | 580.29 |
| | 5 | 2 | 138.72 | 3 | 208.24 | 5 | 347.25 | 8 | 557.42 |
| | 6 | 1 | 80.55 | 3 | 244.95 | 5 | 400.84 | 8 | 645.87 |
| | 8 | 1 | 102.93 | 2 | 205.71 | 4 | 410.97 | 7 | 719.04 |
| | 14 | 1 | 169.66 | 1 | 170.22 | 3 | 509.89 | 5 | 849.04 |
| | 34 | 1 | 393.42 | 1 | 394.49 | 1 | 392.42 | 3 | 1173.16 |
| | 80 | 1 | 899.37 | 1 | 906.18 | 1 | 905.44 | 1 | 905.81 |

The results reported in Table 7.11 suggest that the best solution times are not given by the same stopping criteria for all values of $\sigma$. In fact it appears that for large standard deviations $\sigma = 0.7, 0.9$ the best solution timings are obtained for small *maxitb*. However, for small values of $\sigma = 0.3, 0.5$, the best timings are given by very large *maxitb*.

The case in which *maxitb* is large and $t_{CPU}$ is low corresponds to the situation in which convergence is obtained in one $CG$ iteration. It is clear that in this circumstance the bulk of the computational work is done by the preconditioner ($bSGS$) and very little by the main solver ($CG$). Given that the preconditioner should only serve as a means to improve the conditioning of the system matrix, the results showing just one CG iteration are not considered in the following analysis. On the other hand, this aspect reveals that an independent Gauss-Seidel (symmetric or not) solver could be a very efficient alternative to Krylov subspace iterative schemes. In §7.4.3 results obtained using Gauss-Seidel solvers are reported for all test problems considered in this Chapter.



(a) $d = 4$

(b) $d = 6$

Figure 7.6: Performance analysis of CG preconditioned with $\mathcal{P}_{bSGS}$ for Test Problem 2

Excluding the data associated with one CG iteration, Table 7.4 shows that, in general, a small number of internal iterations for the $\mathcal{P}_{bSGS}$ preconditioner are sufficient to achieve the best performance for all values of standard deviation considered for this test problem. However it is only for $\sigma = 0.9$ ($d = 4, 6$) and $\sigma = 0.7$ ($d = 6$), that the best performance is achieved using $maxitb = 1$. For $\sigma = 0.5$ ($d = 4, 6$) and $\sigma = 0.7$ ($d = 4$), the best performance is given by $maxitb = 2$, and for $\sigma = 0.3$ ($d = 4, 6$), for $maxitb = 3$.

Figures 7.6a and 7.6b show CG iterations versus CPU times for $maxitb = 1, 2, 3, 4, 5, 6$ for $d = 4$ and $d = 6$, respectively. The figures highlight that there is a clear linear relationship between the number of CG (preconditioned with $\mathcal{P}_{bSGS}$) iterations, computational time and the standard deviation of the spatial random field for all values of $maxitb$. As for the linear case, both figures clearly show that the best convergence rate is given by $maxitb = 1$ and this is the reason why it was chosen as the optimal stopping criteria for the $\mathcal{P}_bSGS$ preconditioner.

### 7.4.3 Gauss Seidel Solvers

The performance analysis carried out on test problem 2 in the previous section revealed that for small standard deviation ($\sigma = 0.3$ and $\sigma = 0.5$) the Gauss-Seidel algorithm used as standing alone solver could be a valid alternative to Krylov subspace solvers for the solution of SFEM systems with lognormal conductivity coefficient. The same observation was obtained for the linear case (normal or uniform conductivity coefficient) in Chapter 6. In this section we present results obtained by block symmetric Gauss-Seidel solver ($bSGS$) and non symmetric Gauss-Seidel solver ($bGS$). We aim to show in what circumstances Gauss-Seidel solvers are more efficient than Krylov subspace solvers.

As for the linear case, the symmetric Gauss-Seidel solver includes a forward and a backward sweep per iteration and the algorithm is essentially the one used for $\mathcal{P}_{bSGS}$. The non-symmetric case only includes a forward sweep per iteration. In both cases the stopping criteria is determined by the error norm satisfying a specific tolerance.

The considerations on the re-ordering of the block structure of $A$, pointed out in Chapter 6 for the linear case, may not be valid for the nonlinear case. In fact most re-orderings aim at reducing the bandwidth of the coefficient matrix which is irrelevant for the lognormal case given that $A$ is block dense. In our implementation we retain the structure as presented in Figure 7.1 and obtained by the summation of progressive ($i = 1, \dots, N$) Kronecker terms (see (7.3)). This ordering is the most natural as it represents the summation of decreasing modes obtained from the polynomial chaos expansion of the conductivity coefficient (see (7.1)).

As for $CG$, the tolerance for the GS solvers is set to $10^{-8}$. In each table we list iteration count $N_{it}$ and solution times $t_{CPU}$ for both $bSGS$ and $bGS$. Only experiments using $UMFPACK$ to invert the diagonal blocks of $A$ are reported.

### Test Problem 1 - variable $h$

Table 7.12 lists iteration count and solution times for test problem 1. Results from this table are summarised as follow:

1. Gauss-Seidel solvers are also optimal with respect to the discretisation parameter $h$;
2. Both $bSGS$ and $bGS$ are computationally more efficient than CG with either $\mathcal{P}_{bdiag}$ or $\mathcal{P}_{bSGS}$ preconditioners. The improvement is considerably more significant than for the linear case (see §6.2.3);
3. The $bGS$ solver is computationally more efficient than the symmetric implementation.

### Test Problem 2 - variable $\sigma$

Table 7.13 lists iteration counts and timings for test problem 2. The findings of this table are summarised as follows:

1. GS solvers are not optimal with respect to $\sigma$;

Table 7.12: bSGS and bGS iterations and solution timings - Test Problem 1

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| $bSGS$ | $\frac{1}{32}$ | 4 | 0.28 | 4 | 1.24 | 4 | 5.46 |
| | $\frac{1}{64}$ | 4 | 1.38 | 4 | 5.15 | 4 | 18.7 |
| | $\frac{1}{128}$ | 4 | 8.11 | 4 | 24.3 | 4 | 82.39 |
| $bGS$ | $\frac{1}{32}$ | 5 | 0.18 | 6 | 0.93 | 6 | 4.08 |
| | $\frac{1}{64}$ | 6 | 1.05 | 6 | 3.85 | 6 | 14.03 |
| | $\frac{1}{128}$ | 6 | 6.86 | 6 | 18.34 | 6 | 61.77 |
| $d = 6$ | | | | | | | |
| $bSGS$ | $\frac{1}{32}$ | 4 | 0.71 | 4 | 6.09 | 4 | 43.57 |
| | $\frac{1}{64}$ | 4 | 3.23 | 4 | 20.92 | 4 | 136.23 |
| | $\frac{1}{128}$ | 4 | 15.97 | 4 | 92.94 | 4 | 563.08 |
| $bGS$ | $\frac{1}{32}$ | 5 | 0.46 | 6 | 4.56 | 6 | 32.52 |
| | $\frac{1}{64}$ | 6 | 2.6 | 6 | 15.76 | 6 | 101.74 |
| | $\frac{1}{128}$ | 6 | 11.95 | 6 | 68.84 | 6 | 415.43 |

2. $bSGS$ is computationally more efficient than $CG$ preconditioned with $\mathcal{P}_{bSGS}$ only for small standard deviations;

3. Non-symmetric Gauss-Seidel solver ($bGS$) is very efficient for small and moderate standard deviations. However, for large values of $\sigma$ it is outperformed by $CG$ preconditioned with $\mathcal{P}_{bSGS}$;

4. As for the previous case, the $bGS$ solver is consistently more efficient than the symmetric implementation.

### Test problem 3 - discontinuous-isotropic conductivity field

Table 7.14 lists iteration count and timings for test problem 3. Similar observations to the ones highlighted for test problem 2 are derived from the data presented in this table. Furthermore the results show that a discontinuous conductivity field has no negative impact on the performance of Gauss-Seidel solvers. This becomes evident if

Table 7.13: bSGS and bGS iterations and solution timings - Test Problem 2

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 8 | 0.58 | 9 | 2.91 | 9 | 12.6 |
| $bSGS$ | 0.5 | 13 | 0.92 | 16 | 5.2 | 19 | 26.72 |
| | 0.7 | 21 | 1.47 | 30 | 9.68 | 41 | 57.33 |
| | 0.9 | 33 | 2.31 | 54 | 17.45 | 86 | 120.64 |
| | 0.3 | 9 | 0.32 | 11 | 1.77 | 12 | 8.38 |
| $bGS$ | 0.5 | 15 | 0.53 | 20 | 3.23 | 25 | 17.54 |
| | 0.7 | 23 | 0.82 | 36 | 5.83 | 53 | 37.15 |
| | 0.9 | 35 | 1.25 | 65 | 10.42 | 110 | 77.24 |
| $d = 6$ | | | | | | | |
| | 0.3 | 8 | 1.48 | 9 | 14.06 | 9 | 99.5 |
| $bSGS$ | 0.5 | 13 | 2.4 | 17 | 26.6 | 20 | 220.36 |
| | 0.7 | 22 | 4.06 | 31 | 48.47 | 43 | 470.07 |
| | 0.9 | 35 | 6.45 | 59 | 92.13 | 96 | 1057.31 |
| | 0.3 | 9 | 0.84 | 11 | 8.61 | 12 | 66.24 |
| $bGS$ | 0.5 | 15 | 1.39 | 20 | 15.66 | 25 | 138.46 |
| | 0.7 | 23 | 2.12 | 36 | 28.1 | 53 | 291.58 |
| | 0.9 | 35 | 3.23 | 65 | 50.82 | 111 | 610.59 |

we compare $N_{it}$ for $\delta = 0.5$ for this problem with that of test problem 2 (continuous conductivity coefficient) for $\sigma = 0.5$ (which corresponds to $\delta = \frac{0.5}{1}$).

## 7.5 Comparison and Conclusions

In the previous sections a large number of methods have been tested to identify the most efficient solver for the stochastic formulation of the diffusion problem (nonlinear case). To identify the methods which are the most efficient and robust with respect to $h$, $\sigma$ and discontinuous $\mu$, the data presented in the previous tables are summarised in Figures 7.7, 7.8 and 7.9. Only the case for $p = 4$ is considered and $d = 4, 6$. The methods included in the figures are listed below.

Table 7.14: bSGS and bGS iterations and solution timings - Test Problem 3

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *bSGS* | 0.5 | 11 | 0.78 | 13 | 4.2 | 14 | 19.64 |
| | 0.7 | 17 | 1.2 | 23 | 7.41 | 29 | 40.68 |
| | 1.0 | 32 | 2.25 | 53 | 17.09 | 84 | 117.66 |
| | 1.0,0.7,0.5,1.0 | 32 | 2.24 | 53 | 17.07 | 84 | 118.49 |
| *bGS* | 0.5 | 12 | 0.43 | 15 | 2.43 | 17 | 11.97 |
| | 0.7 | 17 | 0.6 | 25 | 4.07 | 33 | 23.32 |
| | 1.0 | 30 | 1.06 | 53 | 8.59 | 87 | 61.46 |
| | 1.0,0.7,0.5,1.0 | 30 | 1.07 | 53 | 8.58 | 87 | 61.11 |
| $d = 6$ | | | | | | | |
| *bSGS* | 0.5 | 11 | 2.04 | 13 | 20.29 | 15 | 178.55 |
| | 0.7 | 18 | 3.33 | 24 | 37.48 | 31 | 343.28 |
| | 1.0 | 35 | 6.47 | 58 | 90.76 | 95 | 1048.38 |
| | 1.0,0.7,0.5,1.0 | 35 | 6.45 | 58 | 90.82 | 95 | 1049.12 |
| *bGS* | 0.5 | 12 | 1.12 | 15 | 11.73 | 17 | 94.08 |
| | 0.7 | 17 | 1.58 | 25 | 19.58 | 34 | 187.85 |
| | 1.0 | 30 | 2.79 | 53 | 41.42 | 96 | 529.96 |
| | 1.0,0.7,0.5,1.0 | 30 | 2.79 | 53 | 41.48 | 96 | 533.87 |

1. *CG* with $\mathcal{P}_{bdiag}$ (*AMG*)

2. *CG* with $\mathcal{P}_{bdiag}$ (*UMFPACK*)

3. *CG* with $\mathcal{P}_{mean}$ (*AMG*)

4. *CG* with $\mathcal{P}_{mean}$ (*UMFPACK*)

5. *CG* with $\mathcal{P}_{bSGS}$ (*UMFPACK*)

6. *bSGS* (*UMFPACK*)

7. *bGS* (*UMFPACK*)

Note that for the *AMG* case the time required to construct the grids and smoother for the approximation is added to the solution times. The *UMFPACK* case does not require any set-up time.

Figure 7.7 shows the block Gauss-Seidel solvers (both *bGS* and *bSGS*) are the most efficient for all discretisations levels. Among the *CG* solvers the one preconditioned with $\mathcal{P}_{bSGS}$ is the one that performs better both in terms number of iterations and computational time.

Similarly to the linear case, Figures 7.8 and 7.9 show that the conjugate gradient solver preconditioned with $\mathcal{P}_{bSGS}$ is the most efficient method for problems with medium / large standard deviation and discontinuous conductivity. Gauss-Seidel solvers also perform well in these circumstances and for small $\sigma$ they are in fact the best-performing methods.

Mean-based preconditioners are, in general, not robust and efficient for SFEM with lognormal distributions. There is very little difference in terms of performance between the *AMG* and *UMFPACK* versions of the preconditioner.

The outcome of this analysis reveals that CG preconditioned with $\mathcal{P}_{bSGS}$ performs well in all settings considered in this work and therefore should generally be used for the solution of SFEM with lognormal distributions. Gauss-Seidel solvers represent a valid alternative to Krylov subspace iterative methods.

## 7.6 SMFEM solvers

### 7.6.1 Schur complement preconditioner

This section reports the performance of preconditioned MINRES (cf. Chapter 6). The preconditioner used is the one described in §4.6.2. As usual the Schur complement is computed exactly (using e.g. *UMFPACK*) or approximated using one V-cycle of *AMG* code.

**Test problem 1 - variable $h$**

The settings for this test problem are described in §5.2.1. Table 7.15 reports the size of the stochastic space $P$, the number of Kronecker products $N$ and the total number of unknowns for each level of discretisation. Note that $p_{\mathcal{L}}$ is chosen so that the positive-definiteness of $A$ is guaranteed, i.e. $p_{\mathcal{L}} = 2p_u$.

The size of the problem is the same as for the linear case (see Table 6.12). However, $A$ is denser having non-zero contributions for each block.

Table 7.15: Dimensions of $P$, $N$ and total number of unknowns - SMFEM

| | | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | $P$ | 15 | 35 | 70 |
| | $N$ | 70 | 210 | 495 |
| | $h = \frac{1}{32}$ | $77,760$ | $181,440$ | $362,880$ |
| | $h = \frac{1}{64}$ | $309,120$ | $721,280$ | $1,442,560$ |
| | $h = \frac{1}{128}$ | $1,232,640$ | $2,876,160$ | $5,752,320$ |
| $d = 6$ | | | | |
| | $P$ | 28 | 84 | 210 |
| | $N$ | 210 | 924 | $3,003$ |
| | $h = \frac{1}{32}$ | $145,152$ | $435,456$ | $1,088,640$ |
| | $h = \frac{1}{64}$ | $577,024$ | $1,731,072$ | $4,327,680$ |
| | $h = \frac{1}{128}$ | $2,300,928$ | $6,902,784$ | $17,256,960$ |

Table 7.16 reports MINRES iteration count and timings for test problem 1. The table reports results for experiments carried out using the exact version (*UMFPACK*) of the $\mathcal{P}_{Schur}$ and the approximated version (*AMG*). The set-up times for the problem and the preconditioner are reported in Appendix E (Table E.4). The set-up for the preconditioner, i.e. the CPU cost of constructing the coarse grids for $K_0$, is performed only once.

The results included in Table 7.16 can be summarised as follows:

1. The Schur complement preconditioner is optimal or almost optimal with respect to $h$ and $d$. However, there is a small increase in the number of iterations for increasing $p$;

2. It is more difficult to define which version of the preconditioner is more efficient. This seems to depend not only on $h$ but also on the number of random variables, $d$, used for the underlying Gaussian field. For $d = 6$ the exact version of the preconditioner is more efficient than the AMG version. For $d = 5$ (not shown

in Table 7.16) the timings are almost equivalent. Finally, for $d = 4$ the AMG version is more efficient for fine discretisations and less for coarser ones.

Table 7.16: *MINRES* iterations and solution timings for $\mathcal{P}_{scomp}$ - Test Problem 1

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 43 | 4.92 | 44 | 28.74 | 47 | 179.78 |
| | $\frac{1}{64}$ | 43 | 19.32 | 45 | 85.95 | 47 | 390.46 |
| | $\frac{1}{128}$ | 43 | 109.09 | 45 | 443.3 | 48 | $1,845.48$ |
| *AMG* | $\frac{1}{32}$ | 45 | 5.04 | 48 | 31.07 | 49 | 186.07 |
| | $\frac{1}{64}$ | 45 | 15.51 | 48 | 79.58 | 51 | 401.18 |
| | $\frac{1}{128}$ | 47 | 76.85 | 49 | 381.69 | 51 | $1,746.01$ |
| $d = 6$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 43 | 18.06 | 45 | 362.89 | 47 | $6,372.67$ |
| | $\frac{1}{64}$ | 43 | 56.12 | 45 | 639.56 | 48 | $8,547.29$ |
| | $\frac{1}{128}$ | 43 | 275.82 | 45 | $2,178.21$ | 48 | $19,837.92$ |
| *AMG* | $\frac{1}{32}$ | 45 | 18.39 | 48 | 379.57 | 50 | $6,784.32$ |
| | $\frac{1}{64}$ | 47 | 48.77 | 49 | 641.55 | 52 | $8,953.65$ |
| | $\frac{1}{128}$ | 47 | 214.76 | 49 | $2,124.69$ | 52 | $20,740.76$ |

Note that for $h = \frac{1}{128}$ and $d = 6$ solving the non-linear case (Lognormal field) is sixteen times more expensive than the linear case (Gaussian field) (see Table 6.13).

### Test problem 2 - variable $\sigma$

The settings for test problem 2 are described in §7.4.1. The performance of the Schur complement preconditioner for varying $\sigma$ is reported in Table 7.17. As for the previous case the set-up time for the preconditioner is performed only once. This is reported in Table E.5 together with the set-up timings for the test problem itself.

The results reported in Table 7.17 can be summarised as follows:

1. MINRES performance deteriorates significantly for increasing values of $\sigma$. This is in line with all methods considered in this thesis. However for the non-linear case, the usage of SMFEM becomes impractical. In fact, the experiments show

Table 7.17: *MINRES* iterations and solution timings for $\mathcal{P}_{scomp}$ - Test Problem 2

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 62 | 7.32 | 75 | 49.05 | 86 | 330.55 |
| *UMFPACK* | 0.5 | 92 | 10.42 | 127 | 83.82 | 171 | 660.2 |
| | 0.7 | 136 | 15.71 | 226 | 149.58 | 346 | 1,345.68 |
| | 0.9 | 206 | 23.47 | 397 | 262.34 | 698 | 2,710.47 |
| | 0.3 | 67 | 7.91 | 80 | 52.42 | 93 | 357.58 |
| *AMG* | 0.5 | 97 | 11.13 | 136 | 88.92 | 182 | 700.29 |
| | 0.7 | 145 | 16.73 | 238 | 157.69 | 368 | 1,422.33 |
| | 0.9 | 218 | 25.08 | 420 | 276.29 | 733 | 2,846.35 |
| $d = 6$ | | | | | | | |
| | 0.3 | 63 | 25.94 | 76 | 617.75 | 89 | 12,009.58 |
| *UMFPACK* | 0.5 | 95 | 39.68 | 134 | 1,079.71 | 180 | 24,329.95 |
| | 0.7 | 143 | 59.29 | 237 | 1,902.71 | 370 | 51,069.64 |
| | 0.9 | 216 | 90.11 | 424 | 3,418.43 | 753 | 103,277.29 |
| | 0.3 | 67 | 27.6 | 81 | 659.43 | 94 | 13,927.52 |
| *AMG* | 0.5 | 100 | 41.21 | 141 | 1,140.65 | 190 | 26,151.63 |
| | 0.7 | 150 | 61.97 | 249 | 2,007.98 | 385 | 53,101.14 |
| | 0.9 | 227 | 94.27 | 440 | 3,577.61 | 783 | 108,174.47 |

that for $\sigma = 0.9$ it takes more than 30 hours to solve the stochastic linear system for a very coarse discretisation ($h = \frac{1}{32}$);

2. The performance of the AMG and UMFPACK versions of the Schur complement preconditioner is similar.

Additionally it should be noted that for $\sigma = 0.7$, the CPU cost of solving test problem 2 with lognormal conductivity coefficient is about 170 times larger than using uniformly distributed spatial random fields (see Table 6.14).

### Test problem 3 - discontinuous-isotropic conductivity field

As has already been shown for other methods, the performance of the solver and preconditioners are not affected by spatial discontinuities in the conductivity field.

In fact the timings reported in Table 7.18 are comparable to those reported for the continuous test problem in Table 7.17.

Table 7.18 shows that the solver performance depends on the largest value of $\delta$ included in the domain. So, for example, for the case of variable $\delta$ (different coefficients of variation for the four sub-domains), MINRES performance is fully governed by the largest value of $\delta$, i.e. $\delta = 1.0$. In fact the timings are almost equivalent to the case of constant $\delta = 1.0$ for all sub-domains.

Table 7.18: *MINRES* iterations and solution timings for $\mathcal{P}_{scomp}$ - Test Problem 3

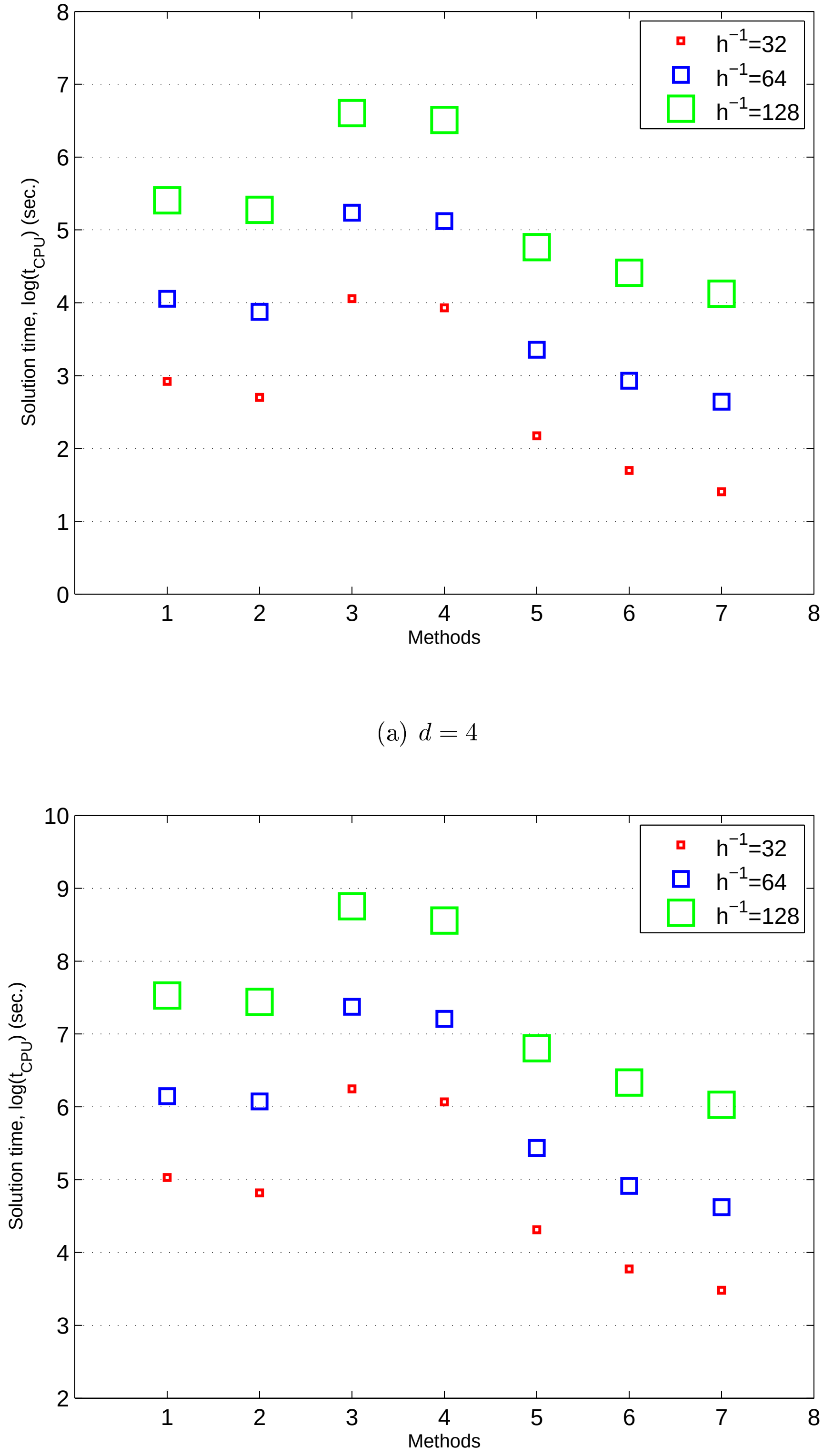
| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) | $N_{it}$ | $t_{CPU}$ (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | 0.5 | 88 | 10.12 | 122 | 80.44 | 161 | 620.97 |
| | 0.7 | 130 | 14.88 | 211 | 139.07 | 316 | 1,226.72 |
| | 1.0 | 235 | 27.15 | 474 | 314.05 | 864 | 3,338.6 |
| | 1.0,0.7,0.5,1.0 | 235 | 27.03 | 473 | 314.9 | 864 | 3,360.14 |
| *AMG* | 0.5 | 92 | 10.44 | 128 | 83.65 | 170 | 651.08 |
| | 0.7 | 136 | 15.55 | 222 | 145.62 | 333 | 1,294.75 |
| | 1.0 | 247 | 28.29 | 499 | 329.87 | 898 | 3,468.6 |
| | 1.0,0.7,0.5,1.0 | 247 | 28.39 | 498 | 327.16 | 898 | 3,484.89 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | 0.5 | 91 | 38.03 | 126 | 1,013.49 | 170 | 23,547.75 |
| | 0.7 | 135 | 56.16 | 223 | 1,814.07 | 341 | 47,604.68 |
| | 1.0 | 250 | 104.13 | 513 | 4,173.31 | 940 | 129,394.1 |
| | 1.0,0.7,0.5,1.0 | 250 | 104.89 | 513 | 4,154.09 | 939 | 131,601.3 |
| *AMG* | 0.5 | 94 | 38.97 | 132 | 1,062.56 | 177 | 24,152.55 |
| | 0.7 | 141 | 58.36 | 231 | 1,870.85 | 355 | 49,131.72 |
| | 1.0 | 259 | 108.41 | 533 | 4,334.59 | 984 | 136,578.2 |
| | 1.0,0.7,0.5,1.0 | 259 | 107.6 | 532 | 4,287.98 | 984 | 136,140.52 |

### 7.6.2 Conclusions

Whilst it was concluded that the performance of MINRES equipped with the Schur complement preconditioner described in (4.57) is acceptable for the solution of the stochastic mixed formulation (linear case), the same cannot be concluded for the non-linear case. The experiments reported in Tables 7.17 and 7.18 show that the CPU cost is too large (30 hours to solve test problem 2 on a coarse mesh, $h = \frac{1}{32}$) for this method to be effectively used with lognormal random fileds.

It becomes apparent that for the non-linear case it is crucial to include information contained in the off-diagonal blocks of the coefficient matrix into the preconditioned system. As already mentioned in Chapter 6, the Kronecker product preconditioner of Ullmann (2008) offers this possibility. Very recently (Powell & Ullmann 2010) extended its implementation to the non-linear case achieving a significant improvement in MINRES CPU cost. The authors also proposed $H(div)$ preconditioning using augmenting schemes which, although being dependent on the choice of the augmentation parameter, seem to achieve very promising results.

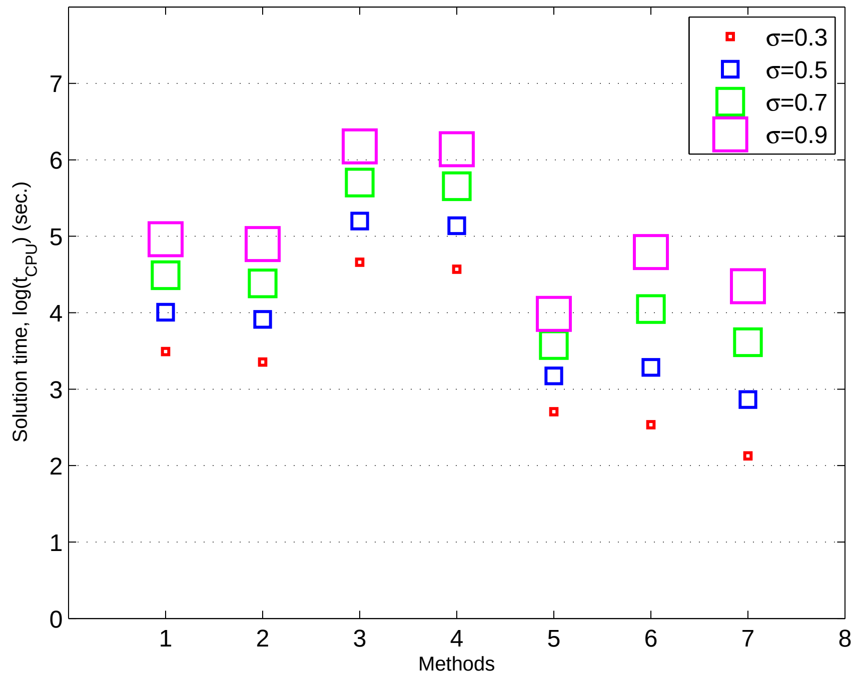
Research in the area of fast iterative solvers for stochastic saddle-point systems is still at the early stages. The ideas proposed in §6.4.2 for the linear case are equally valid for the non-linear case and deserve attention as possible future directions for research.



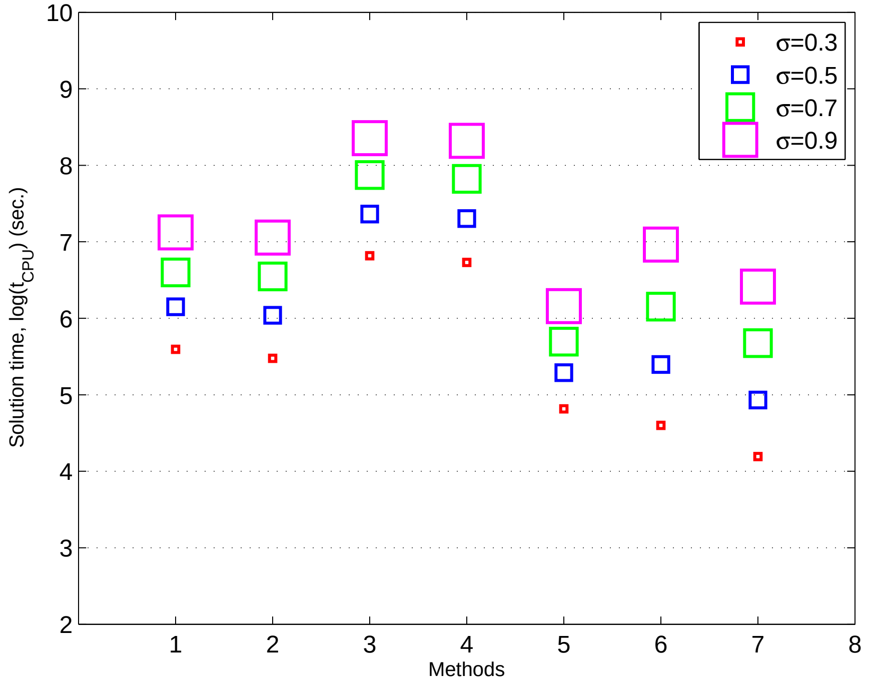
(a) $d = 4$

(b) $d = 6$

Figure 7.7: Comparison of methods for the solution of SFEM for test problem 1

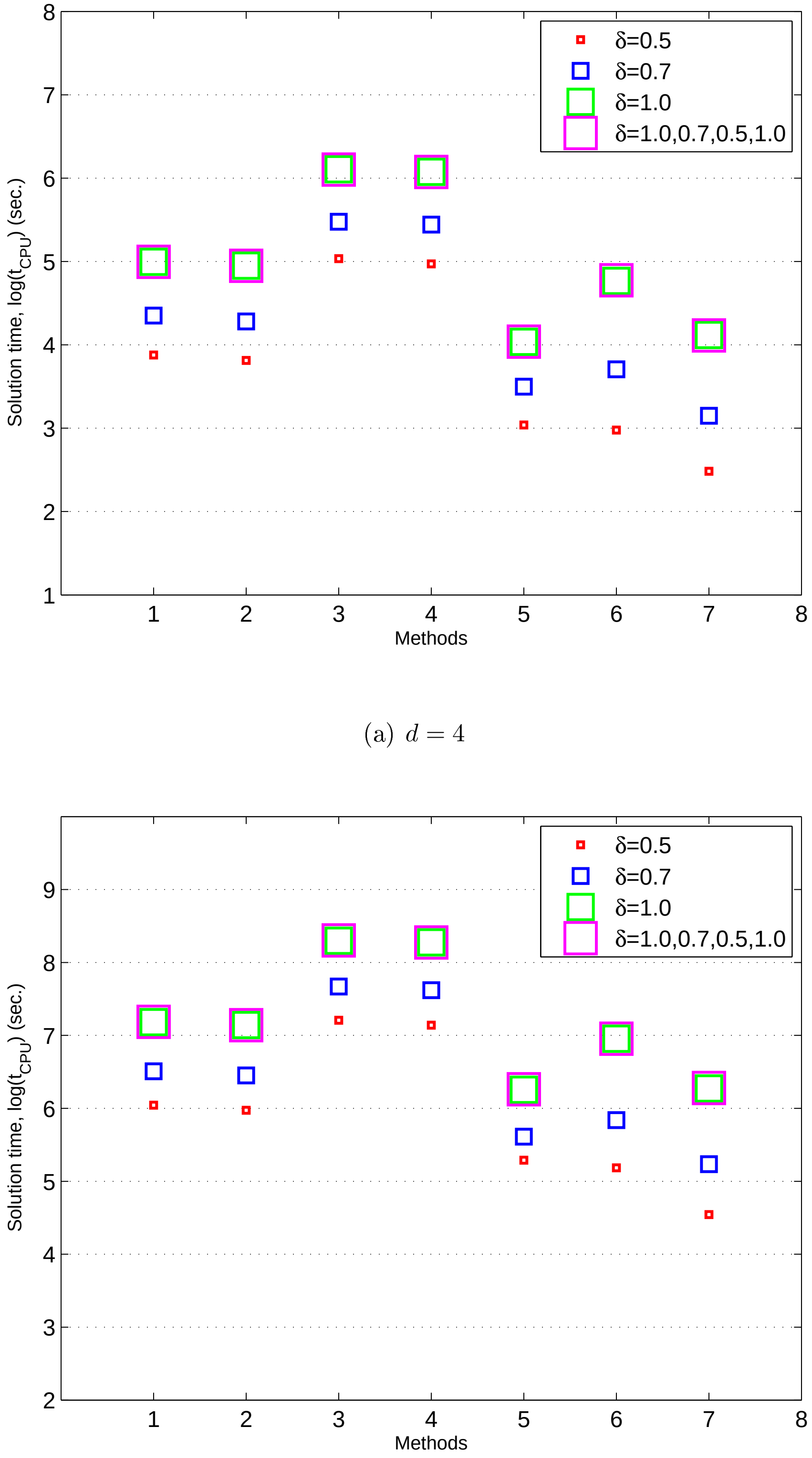


(a) $d = 4$



(b) $d = 6$

Figure 7.8: Comparison of methods for the solution of SFEM for test problem 2



(a) $d = 4$

(b) $d = 6$

Figure 7.9: Comparison of methods for the solution of SFEM for test problem 3

# Chapter 8

# Cardiff Bay Case Study

## 8.1 Introduction

In this chapter the numerical methods presented in the previous chapters are utilised to approximate the distribution of head potential and groundwater velocities in an actual site in the United Kingdom. The location which was selected as a suitable case study is the area surrounding and including the city of Cardiff, capital of Wales. The extent of the site, which covers approximately an area of 15 km$^2$, is shown in Figure 8.1.

Among the reasons this site was selected as a case study are its accessibility and the invaluable support given by the students and lecturers of the MSc course in hydrogeology at the School of Earth and Ocean Sciences in Cardiff University. This site, in fact, has been used over the years for educational purposes in various hydrogeology courses taught in the Master programme. Hence, an extensive understanding of the area was readily available in the department.

More importantly, given the limited extent of the site, the area is unusually rich in field data. This is the result of intense field work carried out in Cardiff and

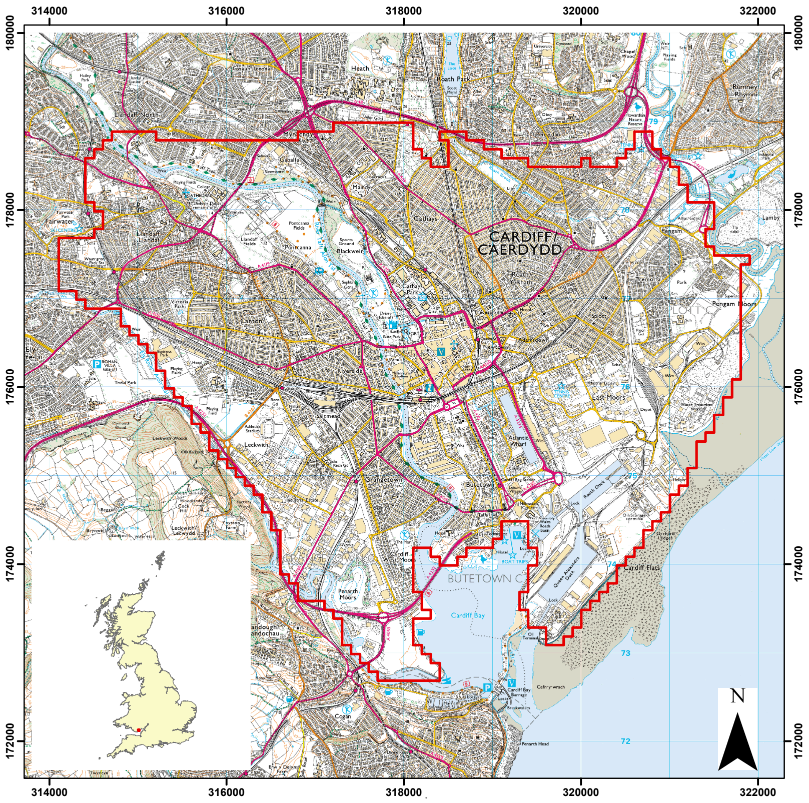


Figure 8.1: Location of study area

surroundings pre- and post-impoundment of the barrage across the mouth of Cardiff Bay, in the early nineties. The threat of a rise in groundwater levels underlying south Cardiff as a consequence of the installation of the barrage and impoundment of a freshwater lake, lead to the construction of a large groundwater monitoring system comprising 236 automated data loggers (Williams 2008). The construction of the barrage itself was supported by an extensive drilling programme from which geological information can be used to construct an accurate geological model for the area.

Furthermore, there exists a groundwater model for the area, developed by Hydrotechnica Ltd. / Entec Ltd., which was used to undertake feasibility studies, preceding the impoundment of the barrage. Although it was not possible to obtain the numerical model, the conceptual characterisation of the study area and a detailed description of the model construction are described in HYDROTECHNICA (1991), Heathcote et al. (1997, 2003).

### 8.1.1 Scope of Groundwater Model Development

The numerical model we aim at developing for the Cardiff Bay area has the primary aim to show that the methodologies presented in the previous chapters can be used for real applications. In particular the aim is to build a groundwater model which can replicate, as close as possible, the distribution and movement of groundwater in Cardiff. The mathematical models used are those described by (2.1) and (2.2) for the deterministic case and (4.1) and (4.2) for the stochastic case.

We will present results for the case in which the model is fully deterministic (transmissivity is known with certainty everywhere in the model domain), and the case in which the transmissivity is described by its mean value and standard deviation (transmissivity is a stochastic process). When the transmissivity is described as a stochastic process the source of uncertainty can be associated with the thickness of the aquifer, the hydraulic conductivity or both. Each of the three cases is analysed.

The case whereby the recharge is a stochastic process is not considered in this work. However the extension of the existing models to the case in which the source term is uncertain is a straightforward matter.

The conceptual model for urban groundwater models, such as the one we intend to develop for Cardiff, can be extremely complex and could include a numerous amount of hydrogeological features which require extensive data analysis and preparation. As our goal is testing and validating the numerical methodologies presented in this thesis, we adopt a simplified version of the conceptual model. Nevertheless, complexity can be increased during subsequent stages of model development. Although the conceptual model is simple, all the important hydrological / hydrogeological features are considered, making it physically sound.

## 8.2 Data Collection and Site Characterisation

### 8.2.1 Location

The site is located in the South Wales region and includes the urban and part of the suburban developments of the city of Cardiff. The study area is bounded by the coast-line and the Cardiff Bay area in the south. The rivers Ely and Rhymney conveniently define the western and eastern boundaries and the A48 approximately defines the northern limit of the site.

The area is crossed by the river Taff discharging, similarly to the Ely river, into the Cardiff Bay area. The river Rhymney discharges into the Severn Estuary (see Figure 8.1).

### 8.2.2 History and Landuse

The city of Cardiff has undergone significant changes in landuse and landscape during the last century. An outstanding review of Cardiff development is given by Gordon et al. (2004). The increase in global demand for Welsh coal resulted in Cardiff growing as a coal port starting from the beginning of the $19^{th}$ century. In less than seventy years the city witnessed the development of seven different docks, starting with Bute West Dock in 1838 and culminating with the construction of Queen Alexandra Dock in 1907.

The rapid development of Cardiff harbour corresponded to a dramatic increase in industrial activities in the the Dockland area. Extensive steel and iron works developed in the East Moors area where the elevation of the ground surface was raised progressively with slag and ash as the works developed and tipping extended out to the present day coastline (Gordon et al. 2004). Other major industries developed around the dockland area including a major gasworks, heavy engineering, paper manufacturing, oil storage terminals and shipbuilding (figures illustrating the development of heavy industries in Cardiff Bay can be found in Gordon et al. (2004), Deane (2010)).

This period of intense industrial growth saw several areas in Cardiff Bay being tipped with domestic refuse, demolition debris and material from construction sites. Remarkable examples are the western area of the city where the River Ely used to be meandering, but between 1954 and 1976, the meanders were cut off and infilled and the area of Ferry Road where the existing gentle hill around 25m high is constructed entirely of domestic refuse.

The dockland area experienced a severe period of decline starting from 1920's due to the fall in the demand of Welsh coal. The whole area of Cardiff Bay soon became disused and derelict. A second major phase of uncontrolled infilling is recorded between 1950's and 1960's, where several of the disused docks were infilled with industrial and domestic waste.

The initial plans to regenerate the dockland area date back to the 1980s. The idea of creating a barrage and impound a freshwater lake at 4.5 mAOD in the bay was conceived in 1993 with the Cardiff Bay Barrage Act. It became immediately evident that the proposed plan could have led to the rise of groundwater levels beneath the city above normal levels therefore causing potential for extensive flooding. As a result provisions were incorporated into the Cardiff Bay Barrage Act including amongst others, a requirement to monitor groundwater before, during and after construction and to consider its control (Williams 2008). The latter was implemented by a series of pumped vertical wells, pumped horizontal collectors and field drains in areas considered particularly at risk from flooding (Williams 2008, Sutton et al. 2004).

The Cardiff Bay Barrage was impounded in November 1999 and completed in March 2000. Since then the areas surrounding the Bay have undergone extensive regeneration and is now home to some prestigious buildings such as the Senyedd, the Olympic Village and the Millennium Centre, as well as large volumes of residential, retail and leisure developments. Today, Cardiff still has a working port with three operational docks. The East Moors are a heavily industrialised area with a large industrial estate, sewerage works and the metal works. A detailed landuse map of modern Cardiff is provided by Deane (2010).

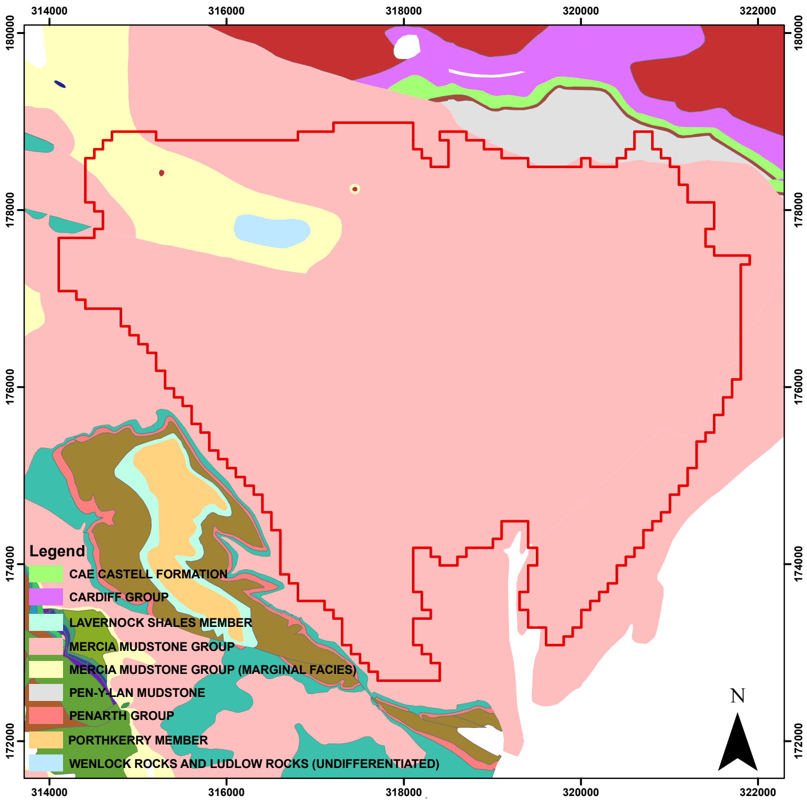
### 8.2.3 Geology

### 8.2.4 Bedrock Geology

The bedrock geology in the Cardiff Bay area is illustrated in Figure 8.2. This comprises the Triassic formation of the Mercia Mudstone group. The sequence is dominated by mudstone but includes a wide range of lithologies ranging from stiff clay to sandstones (Gordon et al. 2004). The vertical and horizontal profiles are highly variable with large differences occurring over small distances.

As pointed out in Edwards (1997), the top surface of the Mercia Mudstone has a prominent topography with distinct features. Its surface is very close to the ground surface along the southern part of the river Ely, and in some instances it forms the western bank of the river. The superficial deposits terminate suddenly against the river bank.

In the northern region of Cardiff the Mercia Mudstone is at or close to outcrop. The continuity of post-glacial superficial deposits is interrupted at several locations



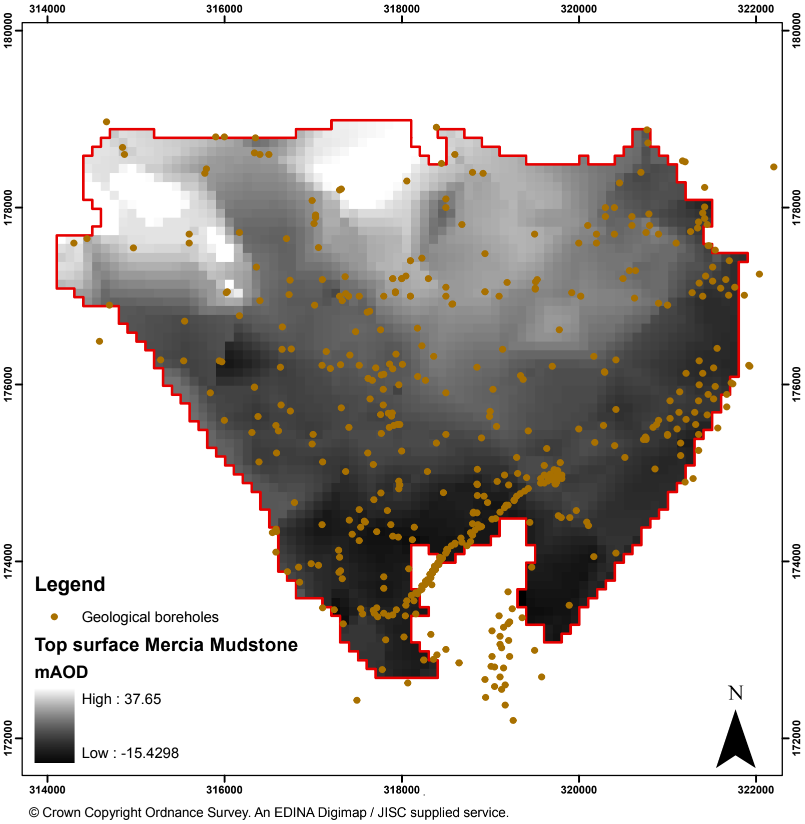
© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.2: Bedrock geology in Cardiff Bay and surroundings

by hills of Triassic and Silurian formations. The area has been also structurally very active with faulting systems having a significant influence on the elevation of the bedrock surface. Further information on the faulting systems existing in the northern region of Cardiff is found in Edwards (1997).

Of the 744 borehole logs analysed, 573 fully penetrate the superficial deposits and reach the bedrock. For the purpose of this work a large database gathering all geological information was created. The top surface of the Mercia Mudstone, obtained by interpolating the elevations collected from the geological logs, is displayed in Figure 8.3. The figure also includes the location of the data upon which the interpolation was made. According to this analysis the elevation of the bedrock ranges from 30 mAOD, in the area of Llandaff and Gabalfa west of the River Taff and the area of

Maindy, to −15 mAOD in the Cardiff Bay region. Its depth from the ground level ranges from < 1 m to 15 m.



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.3: Top surface (mAOD) of Mercia Mudstone and location of geological boreholes

There are several buried valleys in the urban area. The most important are those following the actual river courses of the Ely and Taff.

### Hydraulic Characterisation

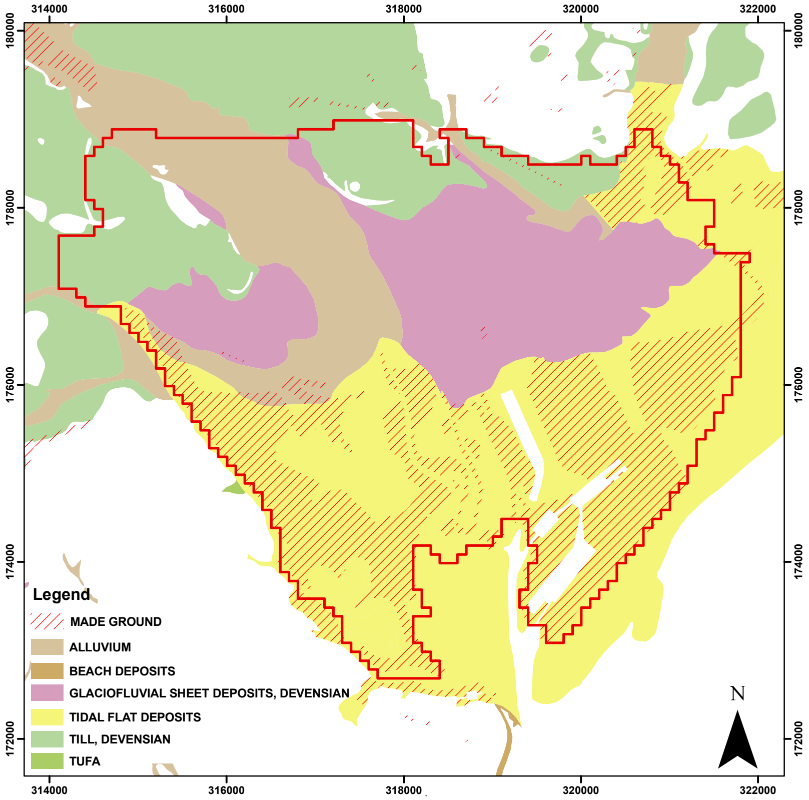
There is evidence of the presence of a sandstone layer in the lower part of the Mercia Mudstone, which has been used for water supply in the past. The upper part, however, is mainly constituted of marl and blocky siltstone, hence acting as an aquiclude, preventing a significant hydraulic connection between the water in the sandstones and the superficial deposits.

Although the water in the lower sandstone can be considered partly isolated, evidence of water circulation in the upper part of the Mercia Mudstone has been reported (HYDROTECHNICA 1991). Few boreholes, in the northern region of the site have recorded water levels and therefore the absence of groundwater movement cannot be discounted completely.

In general, the Mercia Mudstone can be considered as an impermeable base to the groundwater system in the Cardiff area. The evidence of water circulation in the upland areas around the periphery of the site cannot be discarded completely. Hence a relatively small portion of the upper part of the Mercia Mudstone should be considered possessing water yielding properties. From a practical point of view, a number of borehole logs recording weathering and solution cavities in the upper part of the bedrock were identified. These were subsequently used to delineate an area which should be considered as part of the main aquifer present in the site.

### 8.2.5 Superficial Geology

Superficial deposits overlay the top of the Mercia Mudstone and cover most of the study area. Their distribution is illustrated in Figure 8.4 and can be distinguished in river gravel and fluvio-glacial gravel. The latter is a remnant of a fluvioglacial outwash fan dating from the end of the Devensian glaciation (Gordon et al. 2004), covering most of the area, except where the bedrock outcrops and in the area of Cardiff East Moors and Roath Dock (Edwards 1997). The deposits are dense, poorly sorted, sandy gravels with cobbles. Gordon et al. (2004) shows the elevation of the gravel deposits and highlights those areas where thicknesses are greater than 7 m. These cover a significant portion of the site and remarkably locate what appears to be a meltwater channel (roughly corresponding to the actual Taff Valley) infilled with outwash deposits achieving thicknesses comprised between 7 m - 12 m.



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.4: Superficial geology in Cardiff Bay and surroundings

South of the main railway line, the gravel deposits are overlain by a sequence of organic rich clays with subordinate silts, sands and gravels, dominated by soft to very soft clays (Gordon et al. 2004), which are generally referred to as alluvium. It is reported by Allen & Rae (1987) that this lithology was deposited along the margins of the Severn Estuary during the post-Devensian sea level rise. This post-glacial deposit is typical of the superficial geological sequence of the areas surrounding the Severn Estuary. A common feature of the alluvium is the presence of peat, generally located at the base of the transgressive deposits reflecting gradual inundation of the coastal plain (Harris & Turner 2005).

The alluvium is only present in the region south of the Swansea-Newport railway line. In this area it achieves thicknesses of 12 m in places (Sutton et al. 2004) and the upper part of the deposit is generally weathered. The alluvium is cut through by the river channel of the Taff and the docks. However it appears to be intact for the channel of the River Ely.

In addition to estuarine alluvium there are very fine sediments of more recent origin, deposited by the two main rivers in the area. These are poorly to moderately well laminated dark grey clays and silts. Stanley (1995) provided evidence that these are often contaminated with coal, ash, clinker, wood and also marine diatoms.

In the region north of the railway line, there are two distinct glacial tills, one above and one below the fluvio-glacial gravel deposits. Particularly in the northern region the Lower Till deposit is often indistinguishable from the weathered upper part of the Mercia Mudstone. The description reported for the two lithologies are often used in an interchangeable manner in the inspected geological logs. Similarly the tills are occasionally difficult to distinguish from the fluvio-glacial gravels, with which they are partly contemporaneous (Edwards 1997). As illustrated in Figure 8.4, the till deposits increasingly dominate the succession in the northern part of the coastal floodplain.

### Hydraulic Characterisation

The river gravels, fluvio-glacial gravels and sandy tills form a widespread aquifer, which is the most important in the area. Results from a number of aquifer tests (see HYDROTECHNICA (1991) for details) suggest a typical hydraulic conductivity of 50 m/d. This value is characteristic of the central region of the site where the gravel is very thick. Lower values are probably more appropriate in the northern region where the gravel pinches out and in the area around Cardiff East Moors, where the aquifer is formed by thin deposits of till or weathered Mercia Mudstone.

The gravel is confined by the overlain alluvium deposit over most of the area south of the Newport-Swansea railway line. This area is characterised by downward gradients, whereby the water in the made ground moves slowly downwards to the gravel aquifer. This regime is stable in those locations where the alluvium is considerably thick, thus acting as aquiclude. However in those areas where the alluvium is thin or absent there exist upward hydraulic gradients. This aspect was of concern before and after the construction of the barrage and impoundment of a freshwater lake in Cardiff Bay. In fact one of the recognised risks associated with that development was the widespread establishment of upward hydraulic gradients as consequence of the new hydraulic regime. An exception to the confined regime are the lower channel of the Taff and the entrance channels to the Queen Alexandra Dock, at those locations the alluvium has been eroded away therefore exposing the gravel aquifer to tidal influences (Gordon et al. 2004, Heathcote et al. 2003).

In the northern region of the study area the gravel is largely unconfined. The alluvium, in fact, is limited to small areas where localized confining regimes are likely to exist.

Groundwater flow in the alluvium deposit is limited but not completely absent. Hydraulic conductivities have proven difficult to determine. Laboratory testing gave an average value of $1.74 \times 10^{-5}$ m/d, but field pumping tests suggested a range from $2.5 \times 10^{-3}$ to $1.2 \times 10^{-1}$ m/d (Heathcote et al. 2003).

### 8.2.6 Made Ground

The made ground is a very discontinuous lithology overlying the alluvium (see Figure 8.4). It varies from a thin layer of soil / building refuse / weathered rock, particularly in the north of the area, to much more substantial thickness, up to 14 m, at some locations south of Cardiff. Fill materials vary widely in nature and distribution and there are no detailed maps from which the type of the fill can be determined (Edwards 1997).

Gordon et al. (2004) suggests that some fill is directly related to the industry working there at the time, therefore the East Moors have made ground consisting of slag and ash from the iron and steel industry. Other areas were raised using materials from the valley industries such as sandstone quarry waste, colliery spoil, ash and domestic waste. Some of the material came from excavating the docks themselves.

The made ground possesses a very short scale of variability. The lack of information on the type and /or distribution of the filling material makes its characterisation extremely difficult.

### Hydraulic Characterisation

The made ground forms a water-bearing layer, however it is spatially very discontinuous and its yield is very poor. As reported in HYDROTECHNICA (1991), Ltd. (1996), water in the made ground is encountered in many locations at shallow depth. The head potential is almost consistently above those recorded for the gravel aquifer, confirming the existence of downward vertical hydraulic gradients between the two units.

The hydrogeologic regime was stable before the impoundment of the freshwater lake at a constant level of 4.5 m in the bay area. The risk of inverting the hydraulic gradients at some locations, with possible impacts on houses and basements, determined the installations of groundwater control systems in specific areas considered to be vulnerable.

Given the extreme variability of the made ground and the lack of knowledge about its nature and distribution, accurate estimates of its permeability are very difficult to obtain. Various tests have been carried out including trenches and pits. The answers however are very different, sometimes contradictory. The Hydrotechnica report HYDROTECHNICA (1991) suggests that the hydraulic conductivity varies over at least one order of magnitude and values of 1 m/d upwards are recommended.

## 8.3 Hydrology

### 8.3.1 Rainfall

The average annual rainfall in Cardiff is approximately 1076 mm/yr, based on long term average MET Office data. Studies by Ltd. (1996), have revealed that there are significant differences in the amount of rainfall experienced across Cardiff due to distinct physical differences between locations. Cardiff Harbour Authority (CHA) rain gauges located at Bute Park and Cardiff Docks have been recording since 1995 and have highlighted the differences in the pattern of rainfall between the north and south of the study area. Available data indicate that rainfall events at the Docks generally occur at different times and are of differing magnitude from those recorded at Bute Park (Ltd. 1996).

In general rainfall at Bute Park is slightly higher than rainfall at Cardiff Docks. The quantitative analysis carried out by (Deane 2010) seems to suggest that there is an areal variation in rainfall patterns throughout south Cardiff. The average daily rainfall values for both locations were calculated as 2.3 $mm$ for Bute Park and 1.7 $mm$ for Cardiff Docks (Deane 2010). Corresponding rainfall totals for the same period (not specified by the author) were 501.8 $mm$ and 368.3 $mm$, respectively (Deane 2010). Both gauges are located at similar elevations. However, the location of the Docks gauge is very exposed to weather conditions in the Severn Estuary. The prevailing strong winds at this location give rise to a lower recorded rainfall than over the rest of Cardiff.

### 8.3.2 Surface Water Bodies

The main two rivers flowing in the study area are the Rivers Taff and Ely. From an hydrological point of view the two rivers are significantly different.

The River Taff has a significant upland catchment area, approximately 510 km$^2$ upstream of Cardiff Bay, and flows through the steep urban areas of the Welsh valleys. As a result of this, the river produces very high peak flows. Low flow (exceeded 95%) of the time) at Pontypridd is 3.46 m$^3$/s.

The river Ely drains a substantially agricultural lowland catchment which is approximately 163 km$^2$ upstream of Cardiff Bay. Consequently it has much lower peaks. Low flow (exceeded 95%) of the time) at St Fagans is 0.53 m$^3$/s.

According to HYDROTECHNICA (1991), there is evidence that there exists interchange of water between the River Taff and the aquifer at specific locations in the study area. Boreholes water levels showed rapid response to change in level of the River Taff in response to summer rainfall events. Conversely there is less evidence of surface-water / groundwater interaction for the River Ely. In fact, this flows on predominantly low permeable materials in most of the study area.

## 8.4 Conceptual Model and Model Construction

### 8.4.1 Hydrostratigraphic Units

The conceptual model herein presented includes the main elements of a simplified hydrogeological system for the Cardiff area. The top surface of the Mercia Mudstone represents a suitable impermeable base for the system. In those areas where the Mercia Mudstone is close to or at outcrop a minimum thickness to the overlying aquifer is assigned.

In the upland areas around the periphery of the site, the uppermost part of the Mercia Mudstone presents clear evidence of weathering and water circulation. Borehole logs were used to define its extent and thickness. This unit is likely to have hydraulic properties similar to those of the overlying gravel aquifer, hence they are considered as a unique unit.

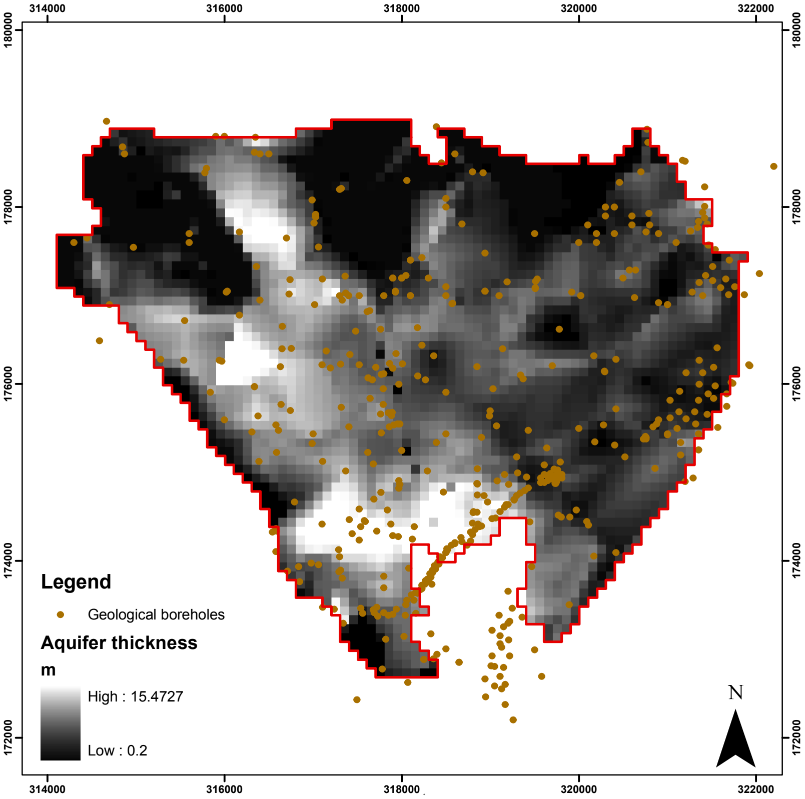
Conceptually the most important hydrostratigraphic unit is the gravel aquifer. This is confined by the Alluvium in the southern part of the site and unconfined in the northern part. The widespread till deposits (mainly located in the region north of the railway line) and river terrace deposits are added to this unit as they possess similar hydraulic properties.

Generally, the low water-bearing strata of the alluvium and made ground are not included in the model as most recent boreholes sampling surveys indicate that these are often dry. Only where clear signs of weathering for the alluvium and where the made ground is in hydraulic contact with the gravel aquifer, are their thicknesses added to the gravel aquifer.

The alluvium and made ground units are hydrogeologically important if, for example, the model was constructed with the scope of investigating the interactions between the upper (made ground) and lower (gravel) aquifers. This is currently outside the scope of this work and the assumption of a single hydrostratigraphic unit seems to be justified. Multi-layered and / or fully three-dimensional models, incorporating all the hydrogeological units in the area are matter of future work.

The thickness of the aquifer used in the model developed in this thesis is illustrated in Figure 8.5 alongside with the kriging interpolator error variance (Deutsch & Journel

1998), Figure 8.6. The latter estimate is based on the location of the data (geological boreholes), the amount of data present in a specified search radius and weighting factors assigned to each data based on their reliability.



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.5: Aquifer thickness in Cardiff Bay

## 8.4.2 Groundwater Levels and Flow Directions

In the Cardiff Bay area there is a large number of observation boreholes at which groundwater levels were recorded during the period preceding and following the construction of the barrage. In this model development only the observed groundwater levels recorded after the impoundment of the freshwater lake in the bay are considered. These, in fact, are representative of the new hydraulic regime established in the region.

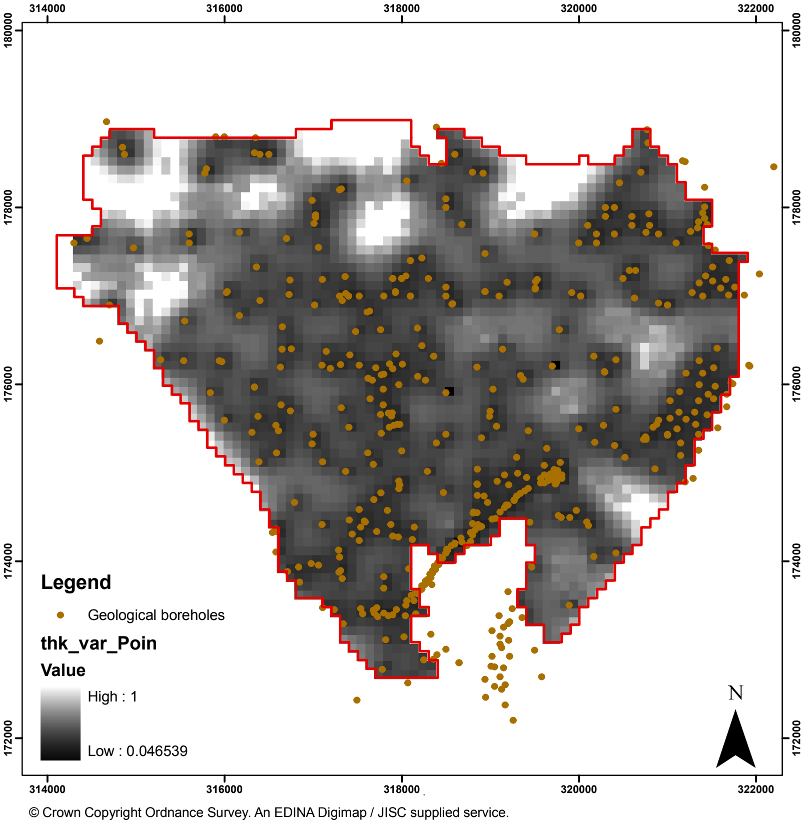


Figure 8.6: Kriging error variance for the aquifer thickness in Cardiff Bay

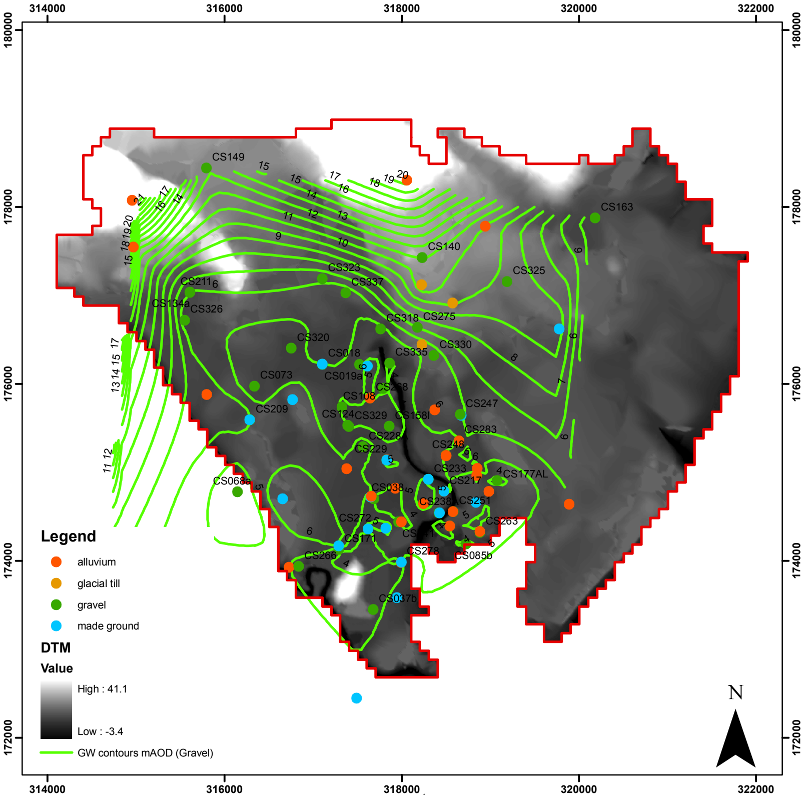
Figure 8.7 illustrates average groundwater contours for the period from July 2003 to September 2004 for the gravel aquifer. The Figure highlights some important features of the groundwater system in the area. The groundwater contours approximately follow the topography, which is also illustrated in Figure 8.7. The latter is clearly influenced by the characteristic profile of the top surface of the Mercia Mudstone. Thus topography and / or top surface of bedrock play an important role on the genaral groundwater flow directions in the area.

The primary groundwater flow direction is north to south-east. The cliffs of bedrock outcropping along the western boundary of the site and posessing a north-west to south-east orientation, play a fundamental role on the groundwater flow dynamics in Cardiff. The water flowing along this direction finally discharges in Cardiff

Bay.

Similarly, the interfluve existing between the Rivers Taff and Ely is a consequence of the bedrock's morphology. This divides the site into two areas and although it is clearly present it cannot be defined precisely due to the scarcity of observation boreholes in the region east of Cardiff. In this work the model area is extended as far as the River Rhymney and one of the objectives of the model output is to identify its location more accurately.

Groundwater flow diverge along the interfluve, partly discharging westerly to Cardiff Bay and partly discharging easterly to the coastline.



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.7: Average observed groundwater levels for the gravel aquifer in Cardiff Bay

The groundwater contours appear to have a very flat gradient in the region corresponding to central Cardiff, with depressions developing at more than one location. In some instances the groundwater levels are almost one metre below the level of the impounded freshwater lake. This is the consequence of the groundwater control systems which abstract water to lower groundwater levels in specific areas of central Cardiff.

### 8.4.3 Material Parameters

Based on the information collected and the study of groundwater levels, the site was divided into 35 zones. An initial hydraulic conductivity value was assigned at each zone with a particular distinction between the northern (north of the Swansea-Newport railway line) and southern regions. In the northen part of the domain the aquifer includes till deposits which are less permeable than gravel deposits. In those areas an hydraulic conductivity of about 5 m/d was assigned.

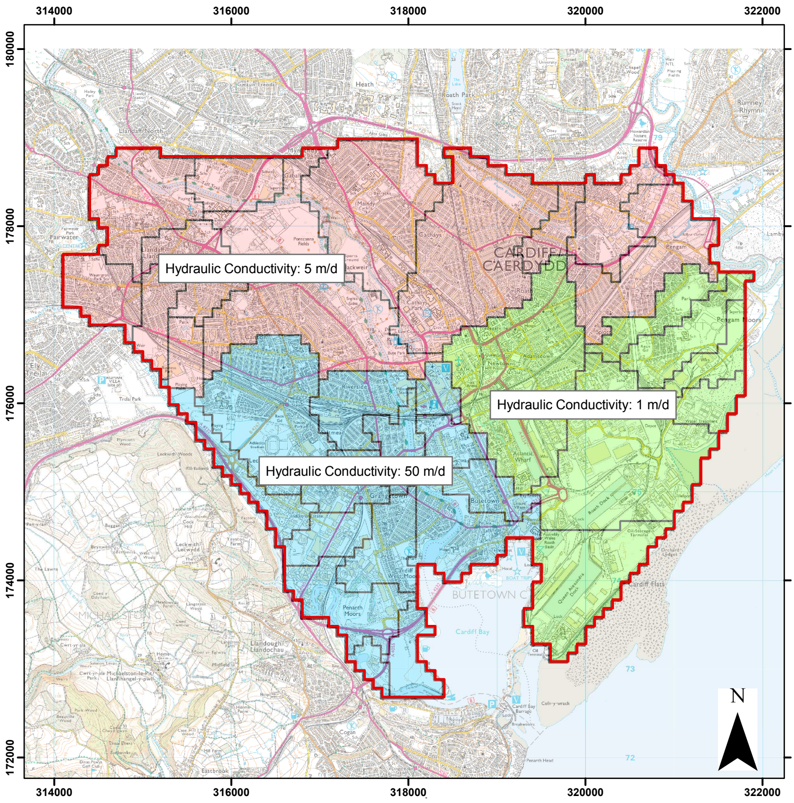
The areas to the east of the Docks present a thin layer of gravel deposit which is actually very often glacial till. Thus an hydraulic conductivity of 1 m/d was initially assigned in those areas.

Elsewhere an hydraulic conductivity of 50 m/d, which is typical of gravel deposits in the area, was assigned.

Figure 8.8, shows the 35 zones of hydraulic conductivity in the site. Note that, although only three values were initially assigned, each of the zones were allowed to change in the parameter estimation process.

### 8.4.4 Potential Recharge

In this section we report estimates of potential recharge to the groundwater system. These approximations were obtained from the water balance calculations reported by HYDROTECHNICA (1991). The analysis distinguishes between permeable and impermeable areas.


314000
316000
318000
320000
322000
180000
178000
176000
174000
172000
Hydraulic Conductivity: 5 m/d
Hydraulic Conductivity: 1 m/d
Hydraulic Conductivity: 50 m/d
CARDIFF
CAERDYDD
BUTETOWN
N

© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.8: Hydraulic conductivity zonation in Cardiff Bay

For the permeable areas the effective rainfall $(actualrainfall-actualevaporation)$ is calculated using a simplified soil moisture model (see HYDROTECHNICA (1991)). The effective rainfall can produce runoff or potential recharge. Runoff will enter either the natural drainage or the sewer system and potentially be removed from the system. The sewer system is very complex in an urban area and the mechanisms whereby the water leaks through the brick walls of the sewers is not considered in this simplified water balance. The potential recharge from rainfall entering the groundwater system is estimated to be around 257 mm/yr.

For impermeable areas it is assumed that no recharge from rainfall occurs and that all rainfall results in evaporation or runoff. However, the leakage from the mains water supply represent another important potential input to the groundwater system. In the city of Cardiff the mains services are located under roads and pavements. Therefore it is assumed that mains water leakage is restricted to impermeable areas.

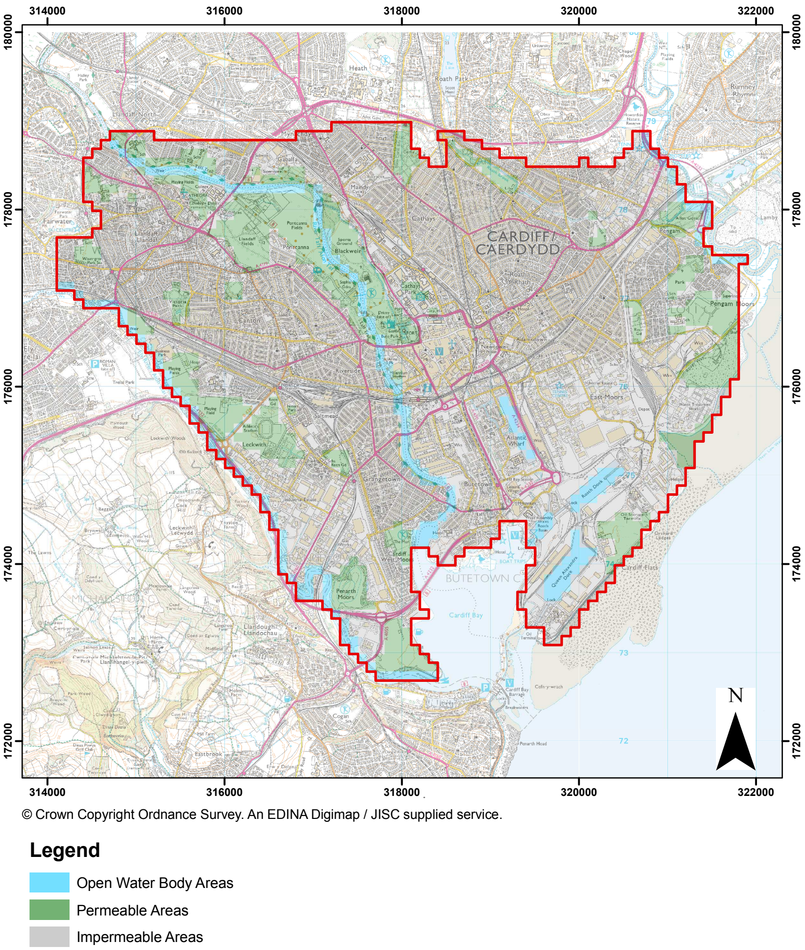
It is estimated that the actual mains water supply entering the model area is approximately $67,415$ m$^3$/d (HYDROTECHNICA 1991). Of this amount, 25% is estimated to leak and become available as potential recharge to the groundwater system and 75% is actually supplied to consumers. Some of the potential recharge could be redirected to the sewers and subsequently removed from the system. Given the obvious difficulties in quantifying the latter mechanism, this is not included in this water balance calculation. The potential recharge from mains water entering the groundwater system is estimated to be around 410 mm/yr.

Finally, in Cardiff there are a number of open water body areas such as rivers and docks. For these areas it is assumed that no recharge from rainfall occurs and that all rainfall results in evaporation and runoff.

Figure 8.9 shows the distribution of the recharge areas in the city of Cardiff. Note that during the calibration process the permeable and impermeable areas are further divided into sub-domains and the recharge values are obtained through the parameter estimation process.

### 8.4.5 Surface-Water Groundwater interaction

The surface-water groundwater interaction has been simulated in a similar fashion to the River Package module in MODFLOW (McDonald & Harbaugh 1988, Harbaugh & McDonald 1996, Harbaugh et al. 2000). This approximation requires the specification of a river stage and a river bed elevation. The river stage data are obtained from the Panorama DTM illustrated in Figure 8.7 and the river bed elevation was chosen to be 1 m below the river stage. Additionally a conductance term is required,



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.9: Distribution of potential recharge in Cardiff Bay

which governs the amount of water which may be transferred from surface-water to groundwater or viceversa. The conductance term is a function of the river bed vertical conductivity, grid cell length, river width and river bed thickness.

Low conductance terms (0.5 m$^2$/d) were set for the Rivers Ely and Rhymney. In fact, as previously explained these rivers mostly flow on the bedrock and alluvium, thus having a low river bed vertical conductivity. Conversely a higher conductance term (10 m$^2$/d) was specified for the River Taff. This, in fact, flows on gravel deposits for most of its length.

### 8.4.6 Boundary Conditions

Figure 8.10 shows the model triangulation and the nodes at which boundary conditions are specified. Only triangular meshes were considered in this chapter, however the extension to quadrilateral / rectangular meshes can be easily carried out.

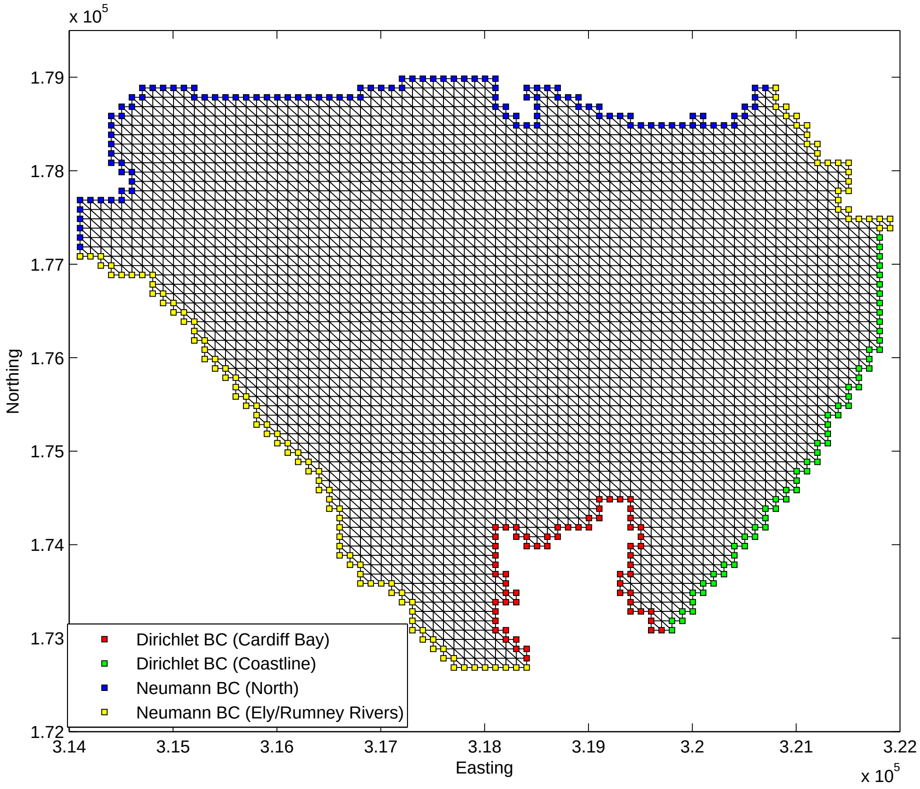


Figure 8.10: Cardiff Bay model - triangulation and location of boundary conditions

The left-hand side boundary of the model correspond to the outcrop of the Mercia Mudstone which parallels the River Ely. Along this boundary the gravel deposits terminates abruptly hence a homogeneous Neumann boundary (no-flow boundary) condition is a very good approximation. The limited surface-groundwater interaction for the River Ely is approximated as a source term (volumetric inflow or outflow) in the adjacent elements to such boundary (see previous section).

The right-hand side boundary corresponding to the River Rhymney has a similar behaviour to the River Ely. This boundary could have been located at the groundwater divide located in the area between the River Taff and Rhymney and visible in the contour plot of groundwater levels (see Figure 8.7). However, given the scarcity of groundwater boreholes in the area east of Cardiff it was not possible to exactly locate the position of such divide. Thus the River Rhymney was chosen as a safer option for the eastern boundary. Note, however, that the approximate location of the groundwater divide can be obtained by the modelling results and the location of the eastern boundary can always be modified in the refining stages of the model development.

The southern boundary corresponds partly with the Cardiff Bay area and partly with coastline between Cardiff and Newport. Since the freshwater lake was impounded at a constant level of 4.5 mAOD, this represents a convenient value for imposing non-homogeneous Dirichlet boundary (constant head boundary) conditions at this location. The coastline boundary approximately follows the mean high water level and thus a value of 0 mAOD represent a good approximation along that boundary.

The northern boundary of the model reflects the bedrock outcrop at some locations, such as in the Llandaff area, and cuts through the gravel aquifer elsewhere. In the model developed by Hydrotechnica Ltd. (HYDROTECHNICA 1991), a non-homogeneous Neumann boundary (specified flow boundary) condition was specified at few cells along the upstream parts of the River Taff. We use the value of 200 $m^3/d$ used in the cited reports but we spread it over all the nodes identifying the northern boundary of the model. Additionally, we subdivide those nodes into different groups depending on their location and estimate the inflow input in the calibration process.

## 8.5 Numerical Model - Deterministic case

Only steady-state simulations are reported in this thesis. This allow us to be consistent with the theoretical discourse reported in Chapters 2 and 4. For the deterministic case the transient development is straightforward but for the stochastic formulation this might be more of a challenge and therefore further investigation is required.

For the deterministic case two types of numerical techniques are implemented, the classic finite element method (FEM) and the mixed finite element method (MFEM). The first uses linear basis functions for the approximation of the potential unknowns (hydraulic head) at the nodal points of the mesh (see Figure 8.10). The velocity solution can be approximated by means of post-processing techniques involving Darcy's law. However for the reasons exposed in 2.1, the velocity approximations obtained in this manner can be significantly erroneous and unphysical.

The latter method uses element piecewise constant basis functions for the approximation of the potential solution and vectorial basis functions for the approximation of the normal fluxes to element edges. Therefore with the MFEM we obtain the potential at each element of the mesh and normal fluxes at each edge of the mesh (see Figure 8.10). The fluxes solution can be subsequently post-processed to obtain the velocity components at the centroid of the elements. Note, however that the latter post-processing does not involve any differentiation and therefore there is no loss of accuracy in the velocity solutions.

For the Cardiff Bay case study the mesh is composed of $3,283$ nodal points, $6,208$ triangular elements and $9,490$ edges. The hydraulic conductivity, potential recharge and river parameters (conductance) are assigned element-wise in both methods. Boundary conditions are assigned nodal-wise for the FEM and edge-wise for the

MFEM.

The discrete linear system obtained from FEM is symmetric and positive definite and can be solved using the conjugate gradient (CG) method. The conditioning of the coefficient matrix can be improved using a preconditioner. Popular choices are approximations of the coefficient matrix by an incomplete Cholesky factorisation or one V-cycle of AMG code. In our simulations we use the second choice.

Conversely the discrete linear system obtained by MFEM is indefinite and therefore CG is not recommended as solver for this type of problems. Other Krylov subspace iterative solvers are suited for indefinite linear systems. In this thesis we use MINRES equipped with the Schur complement preconditoner described in §2.5.3.

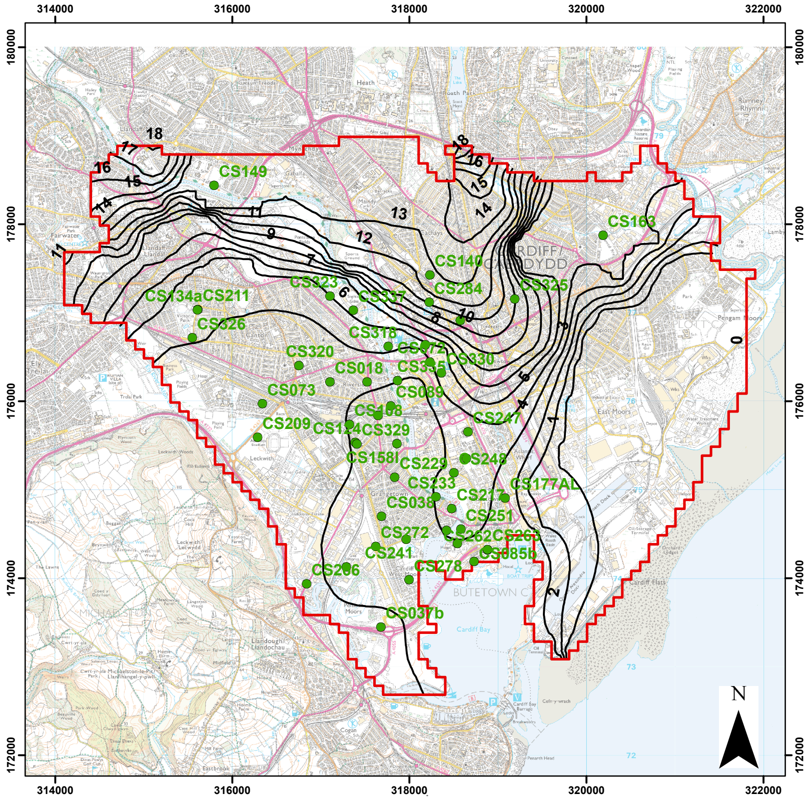
All simulations are run until the solvers have converged. The solvers tolerance is set to $10^{-9}$.

### 8.5.1 Calibration and Model Results

The calibration process is sought as the first step of the model validation process. Validation is the process whereby the numerical model is assessed to be representative of the real situation in an acceptable manner. Obviously this is a mandatory measure if the model was to be used for predictive purposes. The second step in the validation process is the sensitivity analysis. This is dealt with in the next sections.

For the calibration process we interfaced the model with a publicly available parameter estimation software, PEST (Doherty 1994). The parameters selected for the estimation process included the hydraulic conductivity zonation (see Figure 8.8) and the specified flow condition at the northern boundary of the model. These parameters were estimated so that a reasonable match was obtained between observed (overall 47 measurements, see Figure 8.7) and modelled groundwater levels. The fit was considered acceptable if the residual between observed and modelled data was less than one metre.

The distribution of modelled groundwater elevations in the Cardiff area is illustrated in Figure 8.11.



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.11: Hydraulic head solution (mAOD) for the Cardiff Bay model

The FEM approximation for the potential reproduces very well the flat hydraulic gradients in the central region of the study area. Also the steep gradients in the northern region are qualitatively comparable to those obtained by measurements (see Figure 8.11). The general north to south groundwater flow directions with discharge areas clearly identified in Cardiff Bay and the coastline, is well replicated.

The model also reproduces the groundwater divide existing between the Rivers Taff and Rhymney. This seems to be located just east of East Moors, passing through

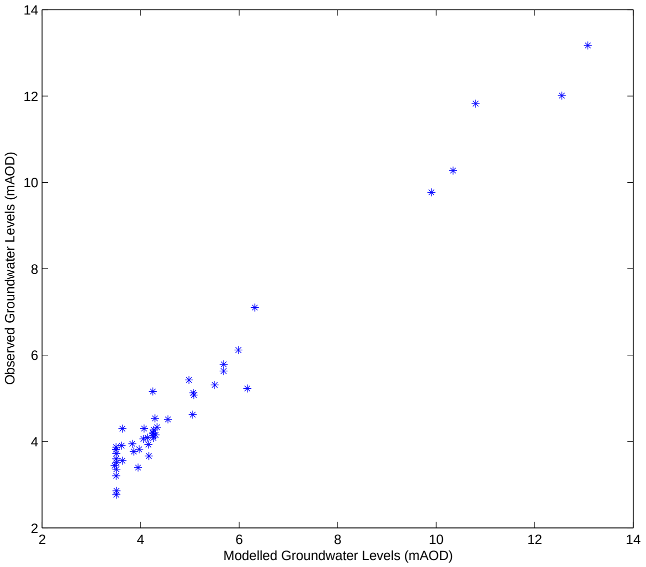


Figure 8.12: Comparison between observed and modelled groundwater levels (mAOD) for the Cardiff Bay model

the suburb areas of Splott and Roath. A model refinement would allow us to define this divide (no-flow boundary) as the eastern boundary of the model.

The calibration plot showing the comparison between modelled and observed data is illustrated in Figure 8.12. Note that if a perfect match existed, the points on the figure would all lie on a 45 degree angle straight line. An indication of the model 'goodness of fit' is given by the sum of squared weighted residuals (SSR), this is equivalent to 6.91 $m^2$ for the calibrated model. The SSR for the uncalibrated model was 214.85 $m^2$. The largest positive residual is 0.9391 m, corresponding to borehole 'CS337' and the largest negative residual is $-1.0245$ m, corresponding to borehole 'CS284'. Both boreholes are located in the proximity of the steep gradients in the northern region of the model. This might indicate that the actual configuration of the conductivity field does not allow one to fully reproduce the variability observed in that area.

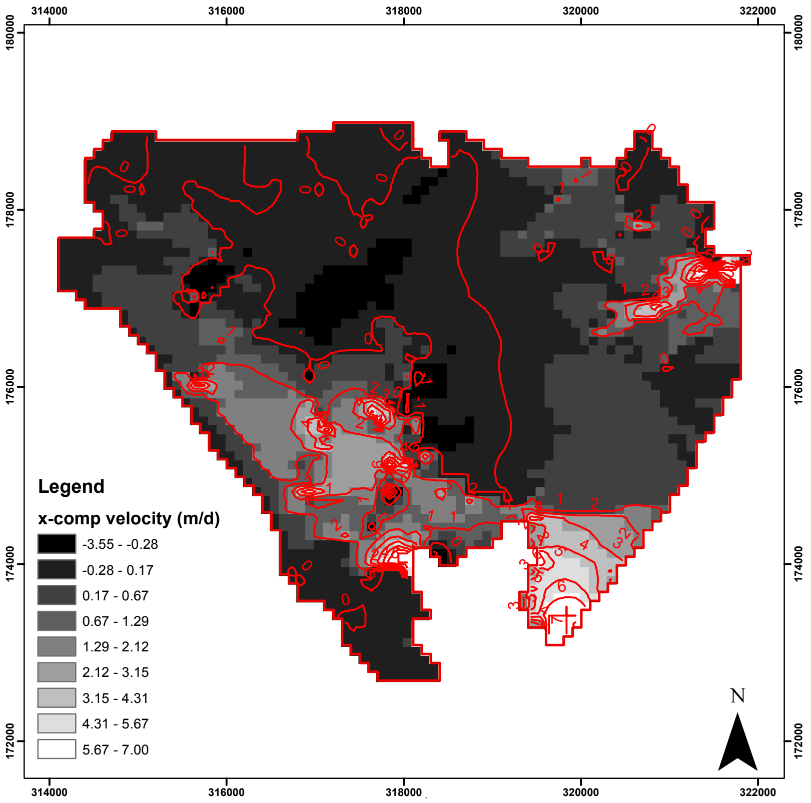
Overall the calibration is very satisfactory, in fact the majority of head residuals

(37 out of 47) are within the range $-0.5 \leq res \geq 0.5$, where $res$ indicates the residual value ($h_{mod} - h_{obs}$) in metres. Only one residual, 'CS284', is outside the acceptable range.

In addition to the hydraulic head approximation, the MFEM approximates the normal fluxes at each edge of mesh. The fluxes can be post-processed to obtain the velocity components at the centroid of each finite element. The MFEM solution for the potential is very similar to the one obtained by FEM, the only difference being that in the latter case the head values are obtained at the nodal points of the mesh while in the former case they are piecewise constant on the finite elements.

The $x$ and $y$ components of the velocity field are pictured in Figures 8.13 and 8.14, respectively. The heterogeneity of the transmissivity field creates a velocity which may appear difficult to interpret at first sight. However some clear features emerge from the numerical results. First of all, the large absolute value of the velocity components agrees with those typical of sand and gravel deposits, which are significant in thickness (exceeding 10 m) and predominantly present in the west region of Cardiff. Secondly groundwater directions largely agree with what is expected from our understanding of the hydrogeologycal system (conceptual model). Positive $x$-component velocities (see Figure 8.13) and negative $y$-component velocities (see Figure 8.14), indicate a westerly and southerly groundwater direction, respectively. This is what is expected in the Riverside and Grangetown areas where the discharge into the freshwater lake of Cadiff Bay is the dominant groundwater mechanism. As already mentioned the large magnitudes of the velocities at these locations are justified by the significant thickness of the aquifer and the large permeability of the gravel.

The northern region (north of the railway line) is characterised by low hydraulic conductivity (the aquifer is often constituted by till deposits) and generally smaller thicknesses. Thus the velocities are small in magnitude and possess a predominant



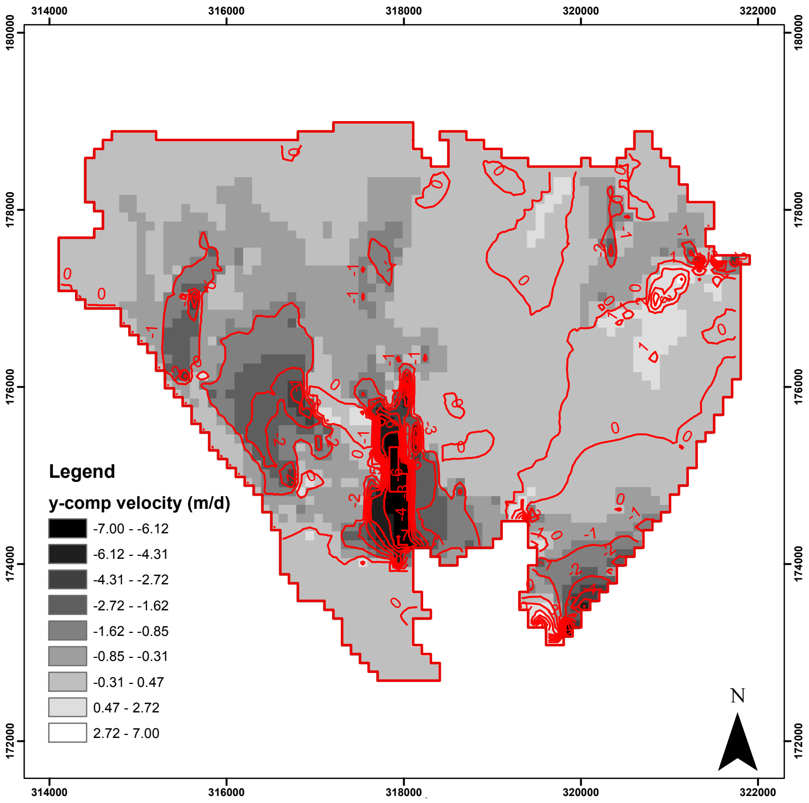
© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.13: $x$-component velocity solution (m/d) in Cardiff Bay

north to south direction.

Thirdly, the groundwater divide, mentioned several times in the conceptual model section is clearly identified by the numerical approximation of groundwater velocities. Its north to south direction coincides with a zero value for the $x$-component of the velocity field. Figure 8.13 clearly shows the groundwater divide existing between the Rivers Taff and Rhymney. A model refinement would allow us to define this feature as the eastern boundary of the model.

Although the models seem to provide physically meaningful solutions for both the hydraulic head and the velocity field, there are some areas in which the numerical approximation should be treated with care. In particular, the strong hydraulic gradient and large easterly directed velocities in the area of Queen Alexandra Dock are caused



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.14: $y$-component velocity solution (m/d) in Cardiff Bay

by the large head jump between the Cardiff Bay and coastline boundaries. This area should be reconsidered at the conceptual level and ways to make this transition less abrupt should be implemented.

In general terms the region to the east of the groundwater divide requires more investigation and additional hydrological-hydrogeological information. Most of the field investigations were in fact carried out in the region west of the groundwater divide. Thus only the model results associated with the latter area should be considered reliable. Future model development would redefine the eastern boundary of the model to correspond with the location of the groundwater divide.

## 8.6 Numerical Model - Stochastic case

The stochastic implementation follows the theory discussed in Chapter 4. However for the reasons discussed in Chapter 7, the conductivity field (and / or transmissivity) is considered to be lognormally distributed.

The numerical methods used for the uncertainty quantification of model predictions belongs to the large family of stochastic Galerkin methods. The Galerkin methods used for the discretisation of the deterministic part are the FEM (with piecewise basis functions on triangular elements) and MFEM (with triangular Raviart-Thomas elements of lowest order). The stochastic part (both, solutions and conductivity coefficient are stochastic processes) is discretised by means of complete multidimensional Hermite polynomials, commonly referred to as polynomial chaos.

A comparison with numerical solutions obtained with Monte Carlo methods for the case study herein discussed is not reported for such a comparison is beyond the scope of this chapter. A comparison, which also served as code validation, between SG and MCM for a number of test problems is presented in Chapters 5 and 7.

For the Cardiff Bay groundwater models presented in §8.5, there are several sources of uncertainty. These are associated with: the conductivity distribution, the thickness of the aquifer, the boundary conditions, the potential recharge to the groundwater system and the rivers parameters (river stage, bottom and conductance). The last two features are incorporated into the mathematical problem as the right-hand side of the PDE to be solved, and they can be thought as a source and /or sink term. The stochastic representation of the source term is not considered in this thesis because, as explained in the introduction, the mathematical challenges are concerned with the stochastic representation of the diffusion or conductivity coefficient.

For this case study the diffusion coefficient corresponds to the transmissivity coefficient, which is the product of the conductivity field and thickness of the aquifer. In a stochastic framework the model input parameters are characterised by the first (mean value) and second (standard deviation) moments. The parameters mean values are directly obtained from the deterministic implementations (see zonation in Figure 8.8), whilst the standard deviation is generally estimated from measurements.

The standard deviation is often interpreted as an indicator of the level of uncertainty associated with a specific parameter. The error associated with the kriging geostatistical interpolator (Deutsch & Journel 1998) associated with the aquifer thickness, illustrated in Figure 8.6, can be successfully used for this purpose. In fact the error (uncertainty) is higher at locations with no measurements and lower at the measurement locations. In the case of Cardiff Bay, there are overall 573 borehole logs at which the aquifer thickness was estimated. These are well-scattered in the study area, thus giving a good representation of the uncertainty associated with the aquifer thickness. Additionally, measurements of thicknesses are less prone to error than other physical parameters such as hydraulic conductivity. This is particularly true in the case of Cardiff Bay where the contact between the top surface of the mudstone and the overlying deposits is well identified.

The uncertainty associated with the conductivity field is more difficult to quantify. Ideally the standard deviation obtained from samples of conductivity measurements for each of the lithologies present in the study area could be used for this purpose. However, conductivity and / or transmissivity data are normally scarce and often erroneous as they are subject to personal interpretation. Furthermore, the measurements are representative at the very local scale and any extrapolation to larger domains is often a conjecture. For the case of Cardiff Bay there are some measurements of hydraulic conductivity, however these are limited and clustered at specific locations. Consequently a representation of the uncertainty associated with the conductivity coefficient based only on measurements is unfeasible and difficult to obtain.

Furthermore, as illustrated in Figure 8.8, the study area has been divided into zones each of which has different hydraulic properties. Therefore we require estimates of the statistics for each of the zones considered in the parameter estimation process. This information is problem dependent and cannot be extrapolated from other studies reported in the public literature.

### 8.6.1 Colored Noise Approach

The first set of simulations consider the hydraulic conductivity as a stochastic process which is spatially correlated whilst the aquifer thickness is a deterministic function depending only on the spatial location.

The hydraulic conductivity is approximated by a discontinuous lognormal spatial random field. A lognormal random field is obtained by an exponential transformation of a Gaussian random field (defined by a Karhunen-Loéve expansion), as explained in §7.2. For a lognormal random field to be used in the context of SG methods, this is subsequently expanded into the polynomial chaos (Ghanem 1999*a*,*b*, Sudret & Der Kiureghian 2000, Ghanem & Spanos 2003, Ullmann 2008).

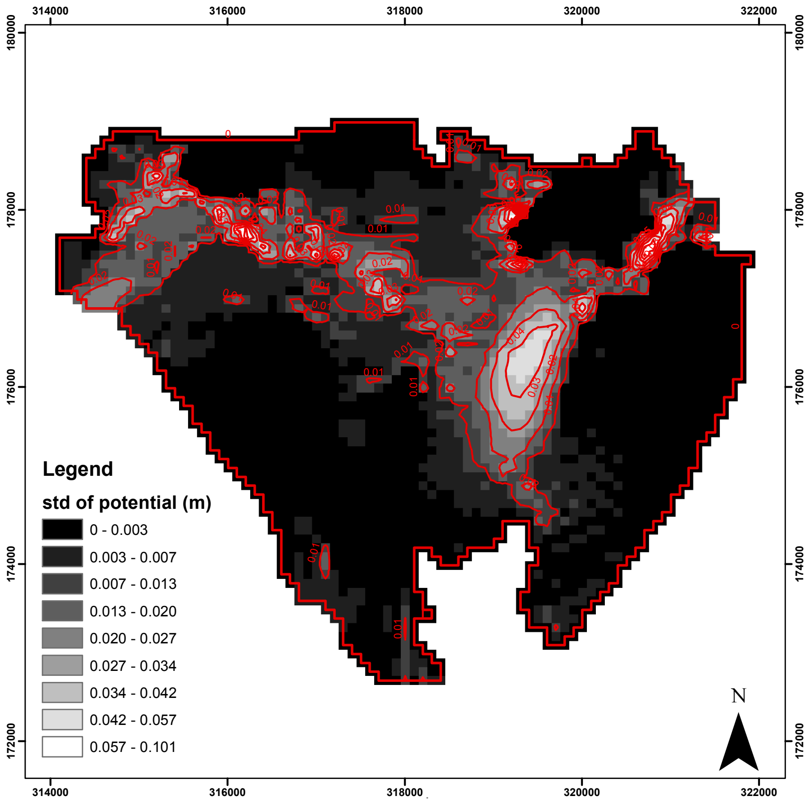
In the case of Cardiff Bay the sub-domains used in the calibration process are used to define the conductivity random field. These sub-domains are of irregular shape as illustrated in Figure 8.8. Providing that the exponential correlation function is a suitable spatial model for the field's spatial variability, the closed form solutions to the eigenvalue problem (4.4) can still be applied to general geometries. It is, in fact, possible to enclose each sub-domain $D_k$, $k = 1, \dots, N_r$ (where $N_r$ is the number of sub-domains), in a square / rectangular shape domain $D_k'$ and solve the eigenvalue problem on the latter. Thus a Karhunen-Loéve expansion is implemented for each of the sub-domains using the calibrated conductivity coefficients as mean values. The standard deviation is obtained by fixing the coefficient of variation to be $\delta = 0.1$.

Given that the KLE point-wise error variance is large at the boundaries of the discretisation domain (Sudret & Der Kiureghian 2000), when compared to other series expansion methods, $D_k^{'}$ is taken larger than the size of the actual region. Thus, if $x_{max}$, $y_{max}$ and $x_{min}$, $y_{min}$ are the maximum and minimum spatial coordinates of $D_k$, respectively, then $D_k^{'}$ is of size $[x_{min} - a_x, x_{max} + a_x] \times [y_{min} - a_y, y_{max} + a_y]$, where $a_x = \frac{x_{max} - x_{min}}{4}$ and $a_y = \frac{y_{max} - y_{min}}{4}$.

The correlation lengths are set to the specific size of each sub-domain and the same number of KLE terms ($d = 4$, i.e four random variables) are retained. As previously explained, each transformed KLE is expanded into polynomial chaos. Therefore we have two polynomial chaos expansions, one used for the conductivity coefficient and one for the solution space. Complete polynomials are used in both cases. However, a maximum degree $p_{\mathcal{L}} = 8$ is used for the coefficient and $p_u = 4$ for the solution (see §7.2).

The mean hydraulic head and velocity components are identical to those illustrated in Figures 8.11, 8.13 and 8.14. The standard deviation associated with the mean hydraulic head solution is illustrated in Figure 8.15. Given that the coefficient of variation is constant, the figure highlights the regions of the model in which the numerical solution is the most sensitive to changes in parameter values. This model output can be compared to the result obtained by implementing a traditional (Monte Carlo based) sensitivity analysis of the conductivity coefficients.

Note that the areas with larger standard deviation correspond to the areas where there is a strong hydraulic gradient (see Figure 8.11). These are the areas in which small changes in the conductivity coefficients determine large changes in the numerical solution. Hence these are the areas where the hydraulic head solution is the most



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

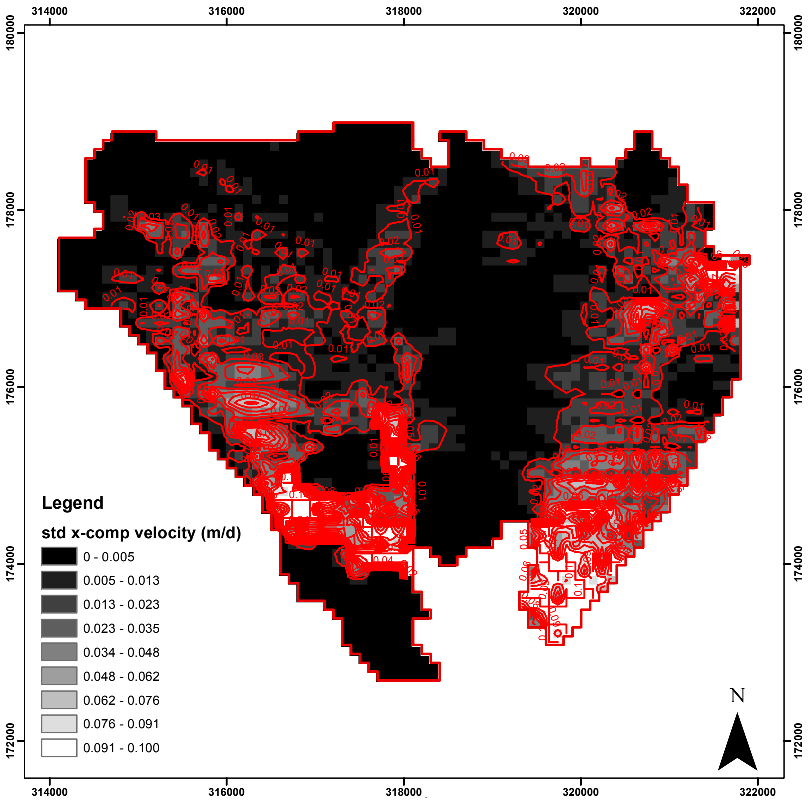
Figure 8.15: Standard deviation of hydraulic head in Cardiff Bay - (randomly) correlated conductivity coefficient

uncertain.

The standard deviation associated with the mean velocity components is illustrated in Figures 8.16 and 8.17. The interpretation of the standard deviation associated with the velocity solution is somewhat less straightforward. However it is evident that the larger uncertainty in model output approximately corresponds to the areas where the velocity components are large in magnitude (see Figures 8.13 and 8.14). These are the areas with large aquifer thickness and /or hydraulic conductivity.

## 8.6.2 White Noise Approach

The white noise approach considers no spatial correlation of the underlying random field. This approach is not normally used to approximate the conductivity



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

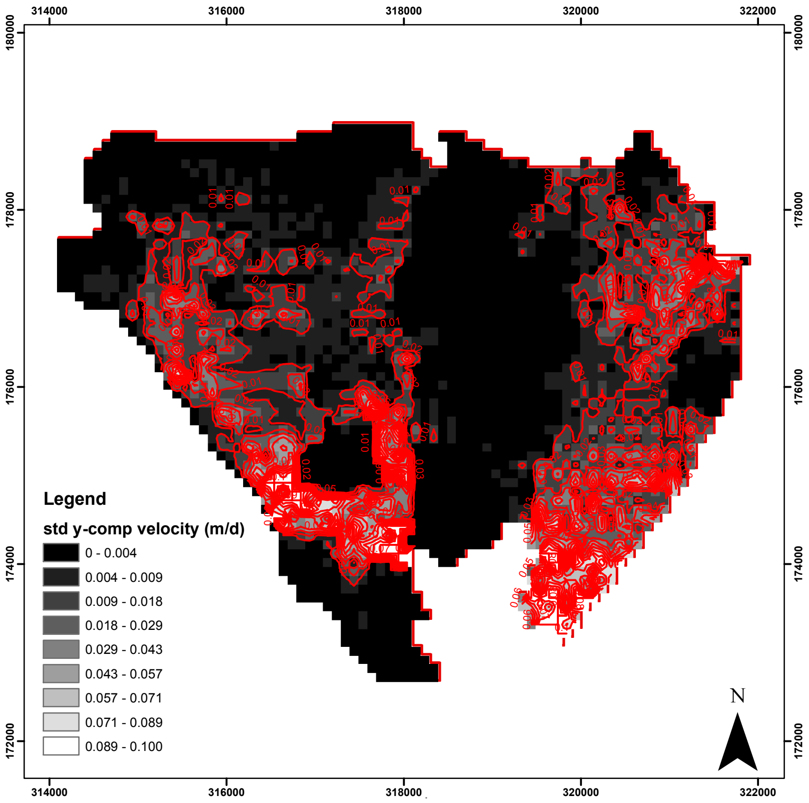
Figure 8.16: Standard deviation of $x$-component of velocity field in Cardiff Bay - (randomly) correlated conductivity coefficient

coefficient which is generally spatial dependent.

For the Cardiff Bay case study the aquifer thickness can be approximated as white noise. In fact the large amount of geological information available allowed us to approximate its spatial variability accurately.

In this section, both the aquifer thickness and the hydraulic conductivity are approximated as white noise. This allow us to quantify the separate contributions of these two parameters to the model output uncertainty.

The white noise approach is implemented setting $d = 1$, thus only the first moment in the KLE is used in the approximation of the spatial random field. A lognormal random variable is defined for the conductivity coefficient and aquifer thickness for each element in the discretised domain (6208 finite elements). The mean values are as



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.17: Standard deviation of $y$-component of velocity field in Cardiff Bay - (randomly) correlated conductivity coefficient

illustrated in Figure 8.5 for the aquifer thickness and as obtained from the calibration process for the conductivity coefficient. The standard deviation for the aquifer thickness is obtained from the kriging interpolation error as illustrated in Figure 8.6. The kriging error was corrected so that the maximum coefficient of variation is equal to one. The standard deviation for the conductivity coefficient was obtained by assigning a constant coefficient of variation, $\delta = 1.0$.

The standard deviation associated with the mean hydraulic head solution is illustrated in Figures 8.18 and 8.19. The first figure shows the hydraulic head standard deviation when the aquifer thickness is a stochastic process and the conductivity coefficient is a deterministic function of space. The second figure shows the opposite situation, i.e. the hydraulic conductivity is a stochastic process and the aquifer thickness is a deterministic function of space.

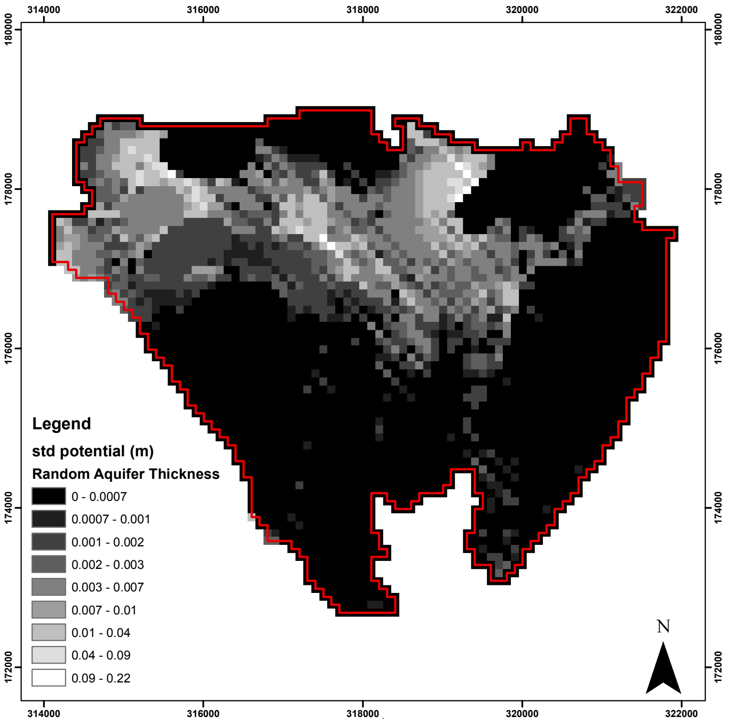
As expected most of the model output uncertainty is due to the uncertainty in the conductivity coefficient. This reflects the fact that the hydraulic conductivity ranges over several order of magnitudes whilst the aquifer thickness is relatively constant. The aquifer thickness uncertainty contribution is limited to small areas in the northern region of the model. This of course is also a reflection of the fact that given the large availability of structural information it was possible to accurately define the aquifer thickness in the study area.

The standard deviation for the velocity components when the aquifer thickness is a stochastic process and the conductivity coefficient is a deterministic process are given in Figures 8.20 and 8.21. The standard deviation for the opposite settings are given in Figures 8.22 and 8.23. Similarly to the hydraulic head, most of the uncertainty in the solution is associated with the conductivity coefficient and only a small amount is due to the aquifer thickness (note that for the figures showing the standard deviation for the aquifer thickness, the plotting scale had to be changed).

## 8.7 Conclusions

In this chapter we have shown that Stochastic Galerkin methods can be effectively used to quantify parameter uncertainty for real-life problems. We used both finite element and mixed finite element techniques to discretise the deterministic part of the variational problem, and multidimensional Hermite polynomials for the discretisation of the probability space.

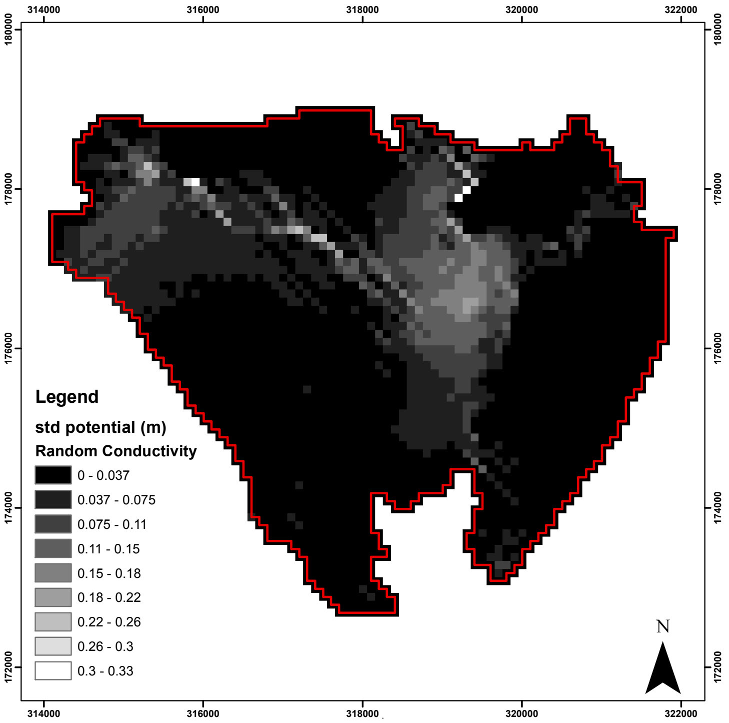
In the current work, we have only taken into account the uncertainty associated with the hydraulic conductivity. However, the stochastic representation of the source term and / or boundary conditions should also be considered. If the various sources



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.18: Standard deviation of hydraulic head in Cardiff Bay - random aquifer thickness

of model uncertainty are described by the same probability distribution, the inclusion of those terms in the stochastic formulation is straightforward. If, however, these are described by different probability distributions, the implementation might be problematic or potentially impossible. To the author's best knowledge studies that consider uncertainty due to probabilistically different parameters have not been reported in the literature. This aspect which might be mathematically difficult to implement and could limit the implementation of the SG methodology, can nevertheless be important from the point of view of applications. In fact, as it has already been discussed, it is generally accepted that parameters, such as hydraulic conductivity, are better approximated by lognormal spatial random fields, but forcing terms, representing for example groundwater recharge, are generally better approximated by uniform spatial or non-spatial random fields. It might be possible that in order to be scientifically grounded these assumptions cannot be relaxed. Thus, in those cases the

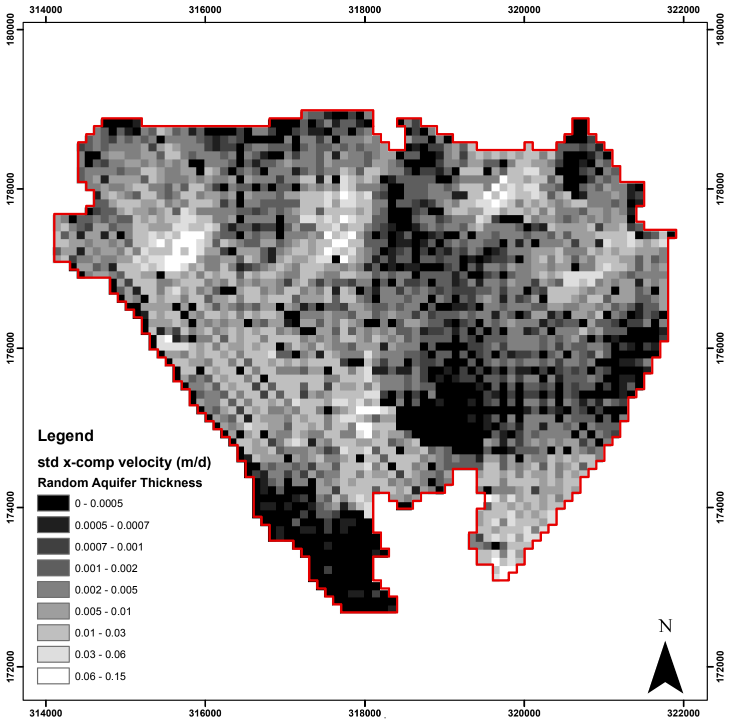


© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.19: Standard deviation of hydraulic head in Cardiff Bay - random conductivity coefficient

use of SG methods should be reconsidered.

The work reported in this chapter has unfolded with a specific logic. A deterministic groundwater model, a simple one in this case, was first developed supported by a thorough conceptual understanding of the study area. The crucial part of this first stage was the calibration process whereby model parameters were adjusted so that numerically computed approximations compared well with observed data. In this work we used the aid of a popular and widely recognised parameter estimation software, PEST (Doherty 1994). This first stage produced a model which integrated not only actual data (collated from field investigation) but which also included experienced knowledge (the subjective understanding that the modeller has developed of the site). In actual fact this model represents the most likely approximation of the site given the available information and understanding. The calibrated parameters are subsequently described in a probabilistic manner using lognormal distributions.

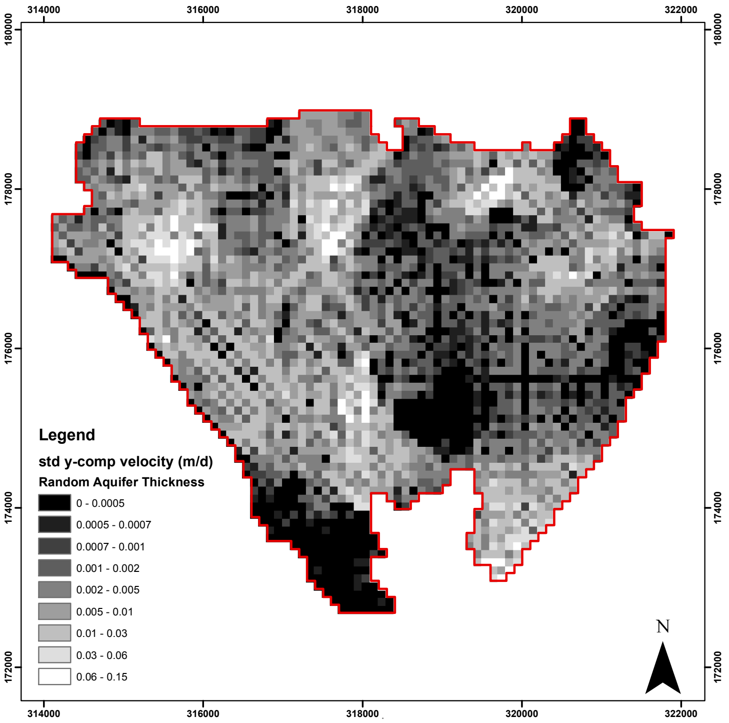


© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.20: Standard deviation of $x$-component of velocity field in Cardiff Bay - random aquifer thickness

Thus the values of the calibrated data set is used to define the mean, and the standard deviation (measure of parameter uncertainty) was given a value determined by adopting a constant coefficient of variation. Although this approach is simplistic it is perfectly reasonable for the scope of this exercise which is primarily methodological. Recently Tonkin & Doherty (2009), Doherty (2010), Herckenrath et al. (2011) proposed a way to use post-calibration information to probabilistically parametrise model input coefficients (conductivity, transmissivity or storativity). In fact, as a result of the calibration process, sensitivities of model results with respect to parameters are computed. Such sensitivities provide an indication of parameters' uncertainty and could be used to statistically characterise the input data used in the stochastic formulation.

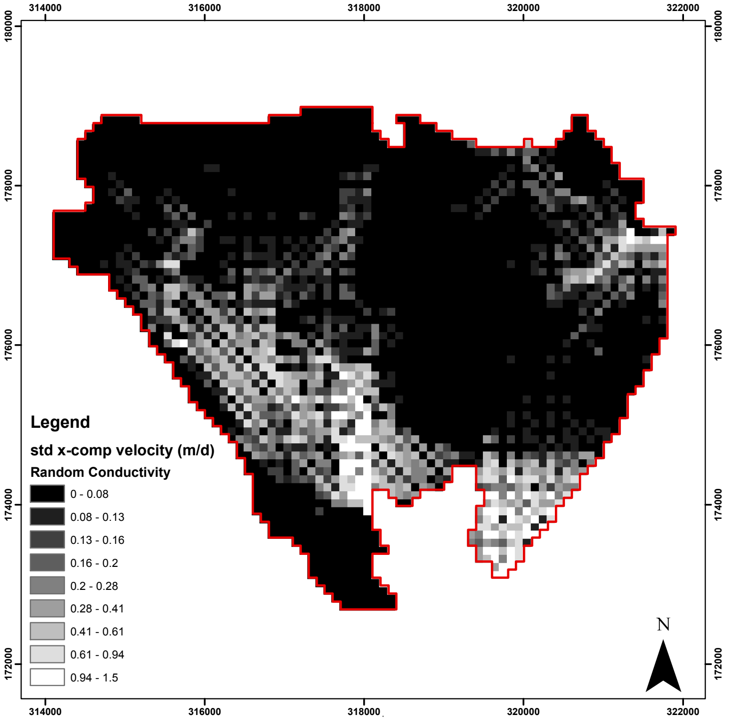
Although approaches to (statistically) characterize model parameters are still being investigated by the scientific community and a general consensus is lacking, the



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

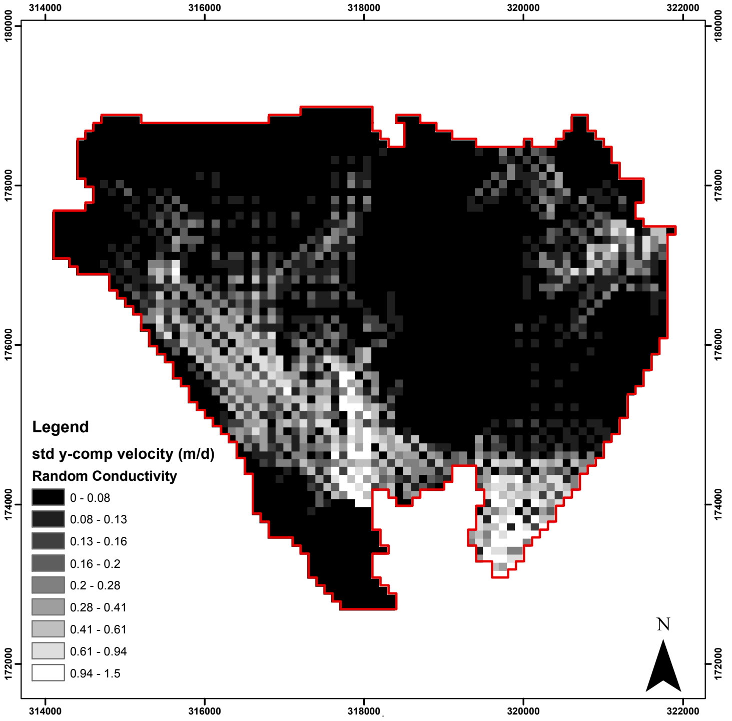
Figure 8.21: Standard deviation of $y$-component of velocity field in Cardiff Bay - random aquifer thickness

author believes that the community should agree that post-calibration is the required starting point for any uncertainty quantification. This is particularly important in the groundwater modelling context where calibrated models hold expert knowledge which is of invaluable importance in not only having a model capable of replicating real-life observations but also of confidently predicting future system behaviour.



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.22: Standard deviation of $x$-component of velocity field in Cardiff Bay - random conductivity coefficient



© Crown Copyright Ordnance Survey. An EDINA Digimap / JISC supplied service.

Figure 8.23: Standard deviation of $y$-component of velocity field in Cardiff Bay - random conductivity coefficient

# Chapter 9

# Conclusions

The work presented in this thesis addresses research questions which are of particular interest to the groundwater modelling community. We first analysed mixed finite element methods which are locally conservative and provide an accurate solution for the velocity field, which is often the variable of primary interest in many groundwater modelling applications. Second, we considered methods for uncertainty quantification - a field which in recent years has emerged as an area of great interest both for academic and industrial communities, as well as for policy-makers. We specifically focused on Stochastic Galerkin methods which have emerged as a valuable alternative to popular Monte Carlo methods. Finally, we applied all the numerical techniques investigated in this thesis to a real-case study in the United Kingdom.

In this concluding chapter we summarise the findings of the study in relation to the underlying research questions and indicate directions for possible future research in relation to each of these.

### Accuracy of MFEM and Computational Comparison with the Hybrid Method

Accuracy of numerically calculated groundwater fluxes and / or velocities has been at the centre of debates for several years. The groundwater modelling community agrees that for some specific applications, such as nuclear waste disposal, the approximations obtained by conventional numerical methods are not accurate enough. Methods that overcome this important limitation are available and the mixed finite element method is one of those.

Our experiments using mixed finite element methods show that discrete error convergence is in line with theoretical results and those reported by other researchers. First order convergence was recorded for the velocity solution on structured and unstructured triangular meshes. This is unaltered for problems with discontinuous, diagonally anisotropic and full-tensor anisotropic coefficients. Second order convergence for the velocity solution is recorded on structured rectangular meshes and a loss of one order convergence is recorded for discontinuous coefficients. As observed in several published articles, the accuracy of the velocity solution deteriorates significantly on unstructured rectangular meshes. The investigation of the approaches that overcome this limitation was outside the scope of this work, but the reader is referred to Younes et al. (2010) for an overview on the matter.

The review of the literature on mixed finite element methods reveals that often the hybridization approach is considered when comparison studies on computational performance are published. The original mixed finite element method is generally discarded on the basis that the discrete linear system of equations is indefinite and larger than the one obtained with the hybrid approach. However, iterative solvers for indefinite systems are available and generally the success of a solver crucially depends on the choice of the preconditioner rather than on the system size. Thus the original mixed finite element method cannot be discarded based only on those considerations.

The results of our experiments show that there is not a unique answer to the proposed research question: is solving the indefinite system computationally more expensive than solving the positive definite system obtained with the hybrid approach? In fact, the performance largely depends on the characteristics of the conductivity coefficient. Thus for problems with isotropic and / or heterogeneous coefficients, on structured or unstructured triangular meshes, solving the indefinite system is the cheapest method. This is also true for anisotropic diagonal coefficients, but only on rectangular meshes. For more general coefficients, i.e. full anisotropic tensors, the experimental results are more complex to summarise. It appears that the AMG implementations generally perform better than all other considered solvers. However, none of these solvers stands out as significantly more effective than others.

### Real-life Application of Stochastic Galerkin Methods

In recent years new methodologies have been developed with the aim of improving the performance of slow converging Monte Carlo methods. In this work we have focused on Stochastic Galerkin methods. We have exposed the limitations of this technology and shown that with suitable assumptions the method can be effectively applied to the groundwater modelling context. To the best of our knowledge the Cardiff Bay case study represents one of the first formal attempts to fully quantify uncertainty in a two-dimensional areal groundwater model.

The approach to uncertainty quantification used in this thesis has evolved through the following steps. First, a deterministic groundwater model for the area was obtained and this was calibrated against observed groundwater levels using parameter estimation techniques. Second, the calibrated conductivity coefficient was mathematically described by a discontinuous spatial random field. The mean conductivity for each sub-regions in the physical domain was assigned the value obtained during the calibration process and the standard deviation was obtained by specifying a constant coefficient of variation. Third, the stochastic problem was formulated using stochastic Galerkin methods and solved.

We argue that this procedure represents a good starting point for further enhancement of uncertainty quantification in groundwater modelling applications. In our analysis only one set of conductivity values that yielded an acceptable calibration was considered for each sub-region. However, a range of parameter sets for which the calibration is deemed to be acceptable could be identified, and a stochastic formulation could be implemented on each of those parameter sets. Furthermore, in our analysis, and as it is often done in industrial applications, the geometry of the sub-regions was determined based on informed subjective judgement. However, there might be several possible geometrical arrangements that would provide equally suitable calibration results. A stochastic formulation could thus be implemented on all of those arrangements. Considering a range of conductivity values and a range of sub-region geometries could thus yield a more robust quantification of parameter uncertainty. Note that this is not the same as implementing a Monte Carlo analysis because the set of considered parameters or geometries are only those for which the model is considered (deterministically) calibrated. The current work thus highlighted this kind of extension of the method as one possible future research direction.

It should also be noted that the model currently used for the Cardiff Bay area is based on a simple conceptual model. For the purpose of the current study this was considered acceptable and it represents a solid starting point from which to build complexity and develop further. The extension of the model to three dimensions and / or multilayered systems can be done. However, additional challenges in terms of computational cost and memory requirements are likely to be encountered. This then represents a further future research direction that arises from this work.

### A New Efficient Preconditioner for SFEM Systems

The success of using Stochastic Galerkin methods relies on efficient implementation and fast iterative solvers. The performance assessment of popular mean-based preconditioners revealed that these are, in general, not robust with respect to the conductivity coefficient. Their performance is considered acceptable for the stochastically linear case but serious limitations were encountered for the stochastically non-linear case.

It was evident from the literature and from our analysis that mean-based preconditioners cannot be robust with respect to the conductivity coefficient because they only include, in the preconditioned system, information associated with the mean value of the spatial random field. The mean information is included in the blocks of the leading diagonal of the global stochastic system, whilst oscillations (representing the variability of the spatial random field) about the mean are contained in the off-diagonal blocks. When the latter contributions become important the mean-based preconditioner performs poorly simply because this information is not included in the preconditioned system. For the stochastically non-linear case this situation is exacerbated by the fact that every block of the global system has non-zero entries.

To overcome this important limitation we proposed an alternative preconditioner for SFEM whereby the off-diagonal blocks of the global system are included in the preconditioned system using a block symmetric Gauss-Seidel algorithm. The analysis of the preconditioner performance revealed that for the stochastically linear and non-linear cases a limited number of iterations for the Gauss-Seidel scheme are required when this is used in conjunction with the conjugate gradient method. In fact, in many cases the best CG performance is achieved when only one Symmetric Gauss-Seidel iteration (a forward and backward sweep) is implemented.

The presented computational analysis clearly showed that block Gauss-Seidel algorithms used either as a preconditioner for CG or as stand-alone solvers are more efficient than mean based preconditioners for both the stochastically linear and non-linear cases. For the latter case, the CPU savings are remarkable. We showed that for some of the test cases considered, the stand-alone standard Gauss-Seidel solver is the best performing solver. However, its performance seems to deteriorate at a faster rate for cases with large standard deviation than preconditioned CG. Therefore, we conclude that generally CG equipped with a block symmetric Gauss-Seidel preconditioner should be used to solve SFEM systems for both the stochastically linear and non-linear cases.

Finally, it should be pointed out that for the non-linear case, the Gauss-Seidel preconditioner performance is poor if one V-cycle of AMG code is used to invert the sub-systems. Our experiments reveal that the AMG based preconditioner is significantly less efficient than the UMFPACK based preconditioner. This is not only due to the considerable set-up time required for the AMG case but it also appears that AMG is less efficient on a per-iteration basis. This finding is different to the results reported for the linear case experiments where AMG always outperforms UMFPACK on a per-iteration basis.

### Considerations on Solvers for Systems Obtained by SMFEM

The presented experiments show that MINRES CPU cost required to solve the stochastic formulation of the mixed finite element method is, in general, very large when a Schur complement preconditioner based on mean information is used. For the stochastically linear case the CPU cost is acceptable. However, it is prohibitively too large for the non-linear case.

The review of the literature available on this topic revealed that this is a very new research area and very few studies have been carried out on efficient solvers for the indefinite systems obtained with SMFEM. As for the deterministic case the hybridization approach is also possible for the stochastic case. However, the advantages existing in the deterministic implementation (the velocity matrix being diagonal) are lost in the stochastic counterpart.

The efficient solution of SMFEM is still an open problem. Some promising results have recently been published by Powell & Ullmann (2010). However, their analysis is very complex and the preconditioners used are not easy and / or practical to implement. I believe, that the decoupling of the velocity vector from the pressure vector, originally proposed by Chavent et al. (1984), Chavent & Jaffré (1986) and more recently by Scheichl (2001) could hold the key to the efficient implementation of the stochastic version of mixed finite element methods, opening up an exciting new area of research.

### Outcomes of the research project

The outcomes of this research will be developed into three articles to be published in international peer-reviewed journals.

The first paper, to be submitted to *Computers and Fluids*, will present a computational comparison between traditional MFEM and MHFEM. Essentially the paper will seek to answer the research question posed at the beginning of this work: under which circumstances is solving the indefinite system obtained from MFEM computationally more expensive than solving the positive definite system obtained with the hybrid approach? The paper is a summary of the numerical work presented in Chapter 3 and builds upon the theory presented in Chapter 2.

The second paper, aimed for publication in *International Journal for Numerical Methods in Fluids*, will compare the computational performance of mean-based and Gauss-Seidel preconditioners for the solution of the stochastic formulation of the groundwater flow equations (second order problem, only). The paper will present results reported in Chapters 6 and 7. It will also include the theory discussed in Chapter 4.

The third paper, aimed for publication in *Water Resources Research*, will present numerical results for the Cardiff Bay case study. In addition to presenting a concrete groundwater modelling application, the paper aims to establish a framework for quantification of model uncertainty in groundwater models. The logical approach which develops from a deterministic, calibrated numerical model to a stochastic model will be emphasized. In order to capture the attention of the wider groundwater research community the paper will include examples of multi-layered groundwater and confined / unconfined systems which are further developed from the work presented in Chapter 8.

In addition to the aforementioned papers, the author plans to write a paper on efficient solvers for the stochastic formulation of the mixed method (first order problem). Differently from the other three, this paper requires substantial additional work, part of which involves understanding whether the decoupling of the velocity vector from the pressure vector, proposed by various researchers in deterministic settings, is also feasible in the context of stochastic Galerkin methods.

# Bibliography

Aavatsmark, I. (2002), 'An introduction to multipoint flux approximations for quadrilateral grids', *Comput. Geosci.* **6**, 404–432.

Aavatsmark, I., T., B., Bøe, . & Mannseth, T. (1998*a*), 'Discretization on unstructured grids for inhomogeneous, anisotropic media. Part I: Derivation of the methods', *SIAM J. Sci. Comput* **19**, 1700–1716.

Aavatsmark, I., T., B., Bøe, . & Mannseth, T. (1998*b*), 'Discretization on unstructured grids for inhomogeneous, anisotropic media. Part II: Discussion and numerical results', *SIAM J. Sci. Comput* **19**, 1717–1736.

Aavatsmark, I., T., B. & Mannseth, T. (1998), 'Control-volume discretization methods for 3D quadrilateral grids in inhomogeneous, anisotropic reservoirs', *SPE Journal* **3**, 146–154.

Allen, J. & Rae, J. (1987), 'Late flandrian shoreline oscillations in the Severn Estuary: a geomorphological and stratigraphical reconnaissance', *Philosophical Transactions of the Royal Society* **B315**, 185–230.

Arbogast, T., Dawson, C., Keenan, M., Wheeler, M. & Yotov, I. (1998), 'Enhanced cell-centered finite differences for elliptic equations on general geometry', *SIAM J. Sci. Comput.* **19**, 404–425.

Arbogast, T., Wheeler, M. & Yotov, I. (1997), 'Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences', *SIAM J. Numer. Anal.* **134**, 828–852.

Arbogast, T., Wheeler, M. & Zhang, N. (1996), 'A nonlinear mixed finite element method for a degenerate parabolic equation arising in flow in porous media', *SIAM J. Numer. Anal.* **33**, 1669–1687.

Arnold, D., Boffi, D. & Falk, R. (2005), 'Quadrilateral H(div) finite elements', *SIAM J. Numer. Anal.* **42**(6), 2429–2451.

Arnold, D. N. & Brezzi, F. (1985), 'Mixed and non-conforming finite element methods: Implementation, postprocessing and error estimates', *$M^2AN$ MAth. Model. Numer. Anal.* **19**(1), 7–32.

Babuška, I. & Chatzipantelidis, P. (2002), 'On solving elliptic stochastic partial differential equations', *Comput. Methods Appl. Mech. Engrg.* **191**(37-38), 4093–4122.

Babuška, I., Nobile, F. & Tempone, R. (2007), 'A stochastic collocation method for elliptic partial differential equations with random input data', *SIAM J. Numer. Anal.* **45**(3), 1005–1034.

Babuška, I., Tempone, R. & Zouraris, G. (2004), 'Galerkin finite element approximations of stochastic elliptic partial differential equations', *SIAM J. Numer. Anal.* **42**(2), 800–825.

Bahriawati, C. & Carstensen, C. (2005), 'Three MATLAB implementations of the lowest-order Raviart-Thomas MFEM with a posteriori error control', *Comput. Methods Appl. Math.* **5**(4), 333–361.

Bause, M., Hoffmann, J. & Knabner, P. (2010), 'First-order convergence of multi-point flux approximation on triangular grids and comparison with mixed finite element methods', *Numer. Math.* **116**, 1–29.

Boyle, J., Mihajlovic, M. & Scott, J. (2007), HSL_MI20: an efficient AMG preconditioner, Technical Report RAL-TR-2007-021, University of Manchester.

Boyle, J., Mihajlovic, M. & Scott, J. (2009), 'HSL_MI20: an efficient AMG preconditioner for finite element problems in 3D', *Int. J. Numer. Meth. Eng.* **82**(1), 64–98.

Braess, D. (1992), *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics*, Cambridge University Press.

Braess, D. & Verfürth (1990), 'Multigrid methods for nonconforming finite element methods', *SIAM J. Numer. Anal.* **27**(4), 979–986.

Brenner, S. (1989), 'An optimal-order multigrid method for P1 nonconforming finite elements', *Math. Comp.* **52**(185), 1–15.

Brenner, S. (1992), 'A multigrid algorithm for the lowest-order Raviart-Thomas mixed triangular finite element method', *SIAM J. Numer. Anal.* **29**(3), 647–678.

Brezzi, F., Douglas, J. & Marini, L. (1985), 'Two families of mixed finite elements for second order elliptic problems', *Numer. Math.* **47**, 217–235.

Brezzi, F. & Fortin, M. (1991), *Mixed and Hybrid Finite Element Methods*, Springer-Verlag.

Brezzi, F., Fortin, M. & L.D., M. (2004), Piecewise constant pressures for Darcy law, *in* F. L.P., ed., 'In Finite Elements Methods: 1970 and Beyond', CIMNE: Barcelona.

Briggs, W., Henson, V. E. & McCormick, S. (2000), *A Multigrid Tutorial*, SIAM.

Cai, Z., Jones, J., McCormick, S. & Russell, T. (1997), 'Control-volume mixed finite element methods', *Comput. Geosci* **1**, 289–315.

Chavent, G., Cohen, G., J., J., Dupuy, M. & Ribera, I. (1984), 'Simulation of two-dimensional waterflooding by using mixed finite elements', *SPE Journal* **24**(4), 382–390.

Chavent, G. & Jaffré, J. (1986), *Mathematical Models and Finite Elements for Reservoir Simulation*, North-Holland.

Chavent, G. & Roberts, J. (1991), 'A unified physical presentation of mixed, mixed-hybrid finite elements and standard finite difference approximations for the determination of velocities in waterflow', *Adv. Water Resour.* **14**(6), 329–348.

Chavent, G., Younes, A. & Ackerer, P. (2003), 'On the finite volume reformulation of the mixed finite element method for elliptic and parabolic PDE on triangles', *Comput. Meth. Appl. Mech. Eng.* **192**, 655–682.

Chen, Z. (1996), 'Equivalence between and multigrid algorithms for nonconforming and mixed methods for second-order elliptic problems', *East-West J. Numer. Math.* **4**(1), 1–33.

Cliffe, K., Giles, M., Scheichl, R. & Teckentrup, A. (2011), 'Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients', *Computing and Visualization in Science* .

Cliffe, K., Graham, I., Scheichl, R. & Stals, L. (2000), 'Parallel computation of flow in heterogeneous media modelled by mixed finite elements', *J. Comput. Phys.* **164**(2), 258–282.

Constantine, P. G. (2009), Spectral Methods for Parameterized Matrix Equations, PhD thesis, Stanford University.

Cordes, C. & Kinzelbach, W. (1992), 'Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements', *Water Resour. Res.* **28**(11), 2903–2911.

Crumpton, P., Shaw, G. & Ware, A. (1995), 'Discretisation and multigrid solution of elliptic equations with mixed derivative terms and strongly discontinuous coefficients', *J. Comput. Phys.* **116**, 343–358.

Davis, T. (2004), 'Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method', *ACM T. Math. Software* **30**(2), 196–199.

Davis, T. & Duff, I. (1997), 'An unsymmetric-pattern multifrontal method for sparse LU factorization', *SIAM J. Matrix Anal. A.* **18**(1), 140–158.

Davis, T. & Duff, I. (1999), 'A combined unifrontal / multifrontal method for unsymmetric sparse matrices', *ACM T. Math. Software* **25**(1), 1–19.

de Dreuzy, J.-R., Beaudoin, A. & J., E. (2007), 'Asymptotic dispersion in 2D heterogeneous porous media determined by parallel numerical simulations', *Water Resour. Res.* **43**(W10439).

Deane, J. (2010), 'Hydrological and hydrogeological investigation of the Cardiff Bay area', Unpublished MSc. Thesis. School of Earth Sciences, Cardiff University.

Deb, M., Babuška, I. & Oden, J. (2001), 'Solution of stochastic partial differential equations using Galerkin finite element techniques', *Comput. Methods Appl. Mech. Engrg.* **190**(48), 6359–6372.

Demlow, A. (2002), 'Suboptimal and optimal convergence in mixed finite element methods', *SIAM J. Numer. Anal.* **39**, 1938–1953.

Deutsch, C. & Journel, A. (1998), *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press.

Diersch, H.-J. G. (1996), Interactive, graphics-based finite-element simulation system FEFLOW for modeling groundwater flow, contaminant mass and heat transport processes, Technical Report FEFLOW User's Manual Version 4.50, WASY Gmbh, Berlin.

Dogrul, E. C. & Kadir, T. N. (2006), 'Flow computation and mass balance in Galerkin finite-element groundwater models', *J. Hydr. Engrg.* **132**(11), 1206–1214.

Doherty, J. (1994), Pest, Model-Independent Parameter Estimation, Technical report, Watermark Numerical Computing. http://www.pesthomepage.org/.

Doherty, J. (2010), Methodologies and software for PEST-based model predictive uncertainty analysis, Technical report, Watermark Numerical Computing. http://www.pesthomepage.org/.

Durlofsky, L. (1994), 'Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities', *Water Resour. Res.* **30**(4), 965–973.

Edwards, M. (2002), 'Unstructured, control-volume distributed, full-tensor finite volume schemes with flow based grids', *Comput. Geosci.* **6**(433-452).

Edwards, M. & Pal, M. (2008), 'Positive definite q-families of continuous subcell darcy-flux cvd (mpfa) finite-volume schemes and the mixed finite element method', *Int. J. Numer. Methods Fluids* **57**(355-387).

Edwards, M. & Rogers, C. (1998), 'Finite volume discretization with imposed flux continuity for the general tensor pressure equation', *Comput. Geosci.* **2**(259-290).

Edwards, M. & Zheng, H. (2008), 'A quasi-positive family of continuous darcy-flux finite volume schemes with full pressure support', *J. Comput. Phys.* **227**(9333-9364).

Edwards, M. & Zheng, H. (2010), 'Double-families of quasi-positive darcy-flux approximations with highly anisotropic tensors on structured and unstructured grids', *J. Comput. Phys.* **229**(594-625).

Edwards, M. & Zheng, H. (2011), 'Quasi M-matrix multi-family continuous darcy-flux approximations with full pressure support on structured and unstructured grids in 3-d', *SIAM J. Sci. Comput.* **33**(455-487).

Edwards, R. (1997), 'A review of the hydrogeological studies for the Cardiff Bay Barrage', *Quarterly Journal of Engineering Geology and Hydrogeology* **30**, 49–91.

Eisenstat, S. (1981), 'Efficient implementation of a class of conjugate gradient methods', *SIAM J. Sci. and Stat. Comput.* **2**, 1–4.

Elman, H., Furnival, D. & Powell, C. (2010), 'H(div) preconditioning for a mixed finite element formulation of the diffusion problem with random data', *Math. Comp.* **79**(270), 733–760.

Ernst, O., Powell, C., Silvester, D. & Ullmann, E. (2009), 'Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data', *SIAM J. Sci. Comp.* **31**(2), 1424–1447.

Ewing, R. & Wheeler, M. (1983), Computational aspects of mixed finite element methods, *in* S. R., ed., 'Scientific Computing', IMACS, North-Holland.

Fischer, B. (1996), *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley-Teubner.

Fortin, M. & Glowinski, R. (1983), *Augmented Lagrangian Methods*, North-Holland.

Fraeijs de Veubeke, B. (1965), Displacement and equilibrium models in the finite element method, *in* O. C. Zienkiewicz & G. Holister, eds, 'Stress Analysis', John Wiley and Sons.

Friis, H., Edwards, M. & Mykkeltveit, J. (2008), 'Symmetric positive definite flux-continuous full-tensor finite-volume schemes on unstructured cell centered triangular grids', *SIAM J. Sci. Comput.* **31**(1192-1220).

Furnival, D. G. (2008), Iterative Methods for the Stochastic Diffusion Problem, PhD thesis, University of Maryland.

Gelhar, L. (1983), *Stochastic Subsurface Hydrology*, Prentice-Hall.

Ghanem, R. (1999*a*), 'The nonlinear gaussian spectrum of lognormal stochastic processes and variables', *ASME Journal of Applied Mechanics* **66**(4), 964–973.

Ghanem, R. (1999*b*), 'Stochastic finite elements for heterogeneous media with multiple random non-gaussian properties', *ASME Journal of Applied Mechanics* **125**(1), 26–40.

Ghanem, R. & Dham, S. (1998), 'Stochastic finite element analysis for multiphase flow in heterogeneous porous media', *Transp. Porous Media* **32**, 239–262.

Ghanem, R. G. & Kruger, R. (1996), 'Numerical solution of spectral stochastic finite element systems', *Comput. Methods Appl. Mech. Eng.* **129**, 289–303.

Ghanem, R. & Spanos, P. (2003), *Stochastic Finite Elements*, Dover Publications Inc.

Giles, M. (2008), 'Multilevel Monte Carlo path simulation', *Operations Research* **56**(3), 981–986.

Giles, M. & Waterhouse, B. (2009), 'Multilevel quasi-Monte Carlo path simulation', *Radon Series Comp. Appl. Math.* **8**, 1–18.

Goode, D. J. (1990), 'Particle velocity interpolation in blockcentered finite difference groundwater flow models', *Water Resour. Res.* **26**(5), 925–40.

Gordon, T., Skuse, M. & Statham, I. (2004), 'Urban geology of Cardiff centre and the Bay region', Urban Geology of Wales I, Geological series no.23. National Museum of Wales.

Graham, G., Kuo, F., Nuyens, D., Scheichl, R. & Sloan, I. (2011), 'Quasi-Monte Carlo methods for for elliptic PDEs with random coefficients and applications', *Journal of Computational Physics* **230**, 3668–3694.

Hackbush, W. (2003), *A Multi-Grid Methods and Applications*, Springer-Verlag.

Harbaugh, A., Banta, E., Hill, M. & McDonald, M. (2000), MODFLOW-2000, the U.S. Geological Survey modular ground-water model. user guide to modularization concepts and the ground-water flow process, Technical Report USGS Open-File Report 00-92, Reston, Virginia: USGS.

Harbaugh, A. & McDonald, M. (1996), User's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model, Technical Report USGS Open-File Report 96-485, Reston, Virginia: USGS.

Harris, C. & Turner, M. (2005), 'Gas monitoring in urban boreholes, Cardiff 1995 - 2002', Urban Geology of Wales II, Geological series no.23. National Museum of Wales.

Heathcote, J., Lewis, R., Russell, D. & Soley, R. (1997), 'Cardiff Bay Barrage: investigating groundwater control in a tidal aquifer', *Quarterly Journal of Engineering Geology and Hydrogeology* **30**, 63–77.

Heathcote, J., Lewis, R. & Sutton, J. (2003), 'Groundwater modelling for the Cardiff Bay Barrage, UK - prediction, implementation of engineering works and validation of modelling', *Quarterly Journal of Engineering Geology and Hydrogeology* **36**, 159–172.

Herckenrath, D., Langevin, C. & Doherty, J. (2011), 'Predictive uncertainty analysis of a saltwater intrusion model using null-space Monte Carlo', *Water Resour. Res.* **47**(W05504), doi:10.1029/2010WR009342.

HYDROTECHNICA (1991), Cardiff Bay Barrage - final report on groundwater modelling, Technical report, Hydrotechnica Ltd. (1991), International consultants in water and environment.

Kaasschieter, E. F. (1995), 'Mixed finite elements for accurate particle tracking in saturated groundwater flow', *Adv. Water Resour.* **18**(5), 277–294.

Kaasschieter, E. F. & Huijben, A. J. M. (1992), 'Mixed-hybrid finite elements and streamline computation for the potential flow problem', *Numer. Meth. Part. D. E.* **8**(3), 221–266.

Keese, A. (2004), Numerical Solution of Systems with Stochastic Uncertainties - A General Purpose Framework for Stochastic Finite Elements, PhD thesis, Technische Universität Braunschweig.

Kim, C. (2001), On Iteration and Approximation Methods for Anisotropic Problems, PhD thesis, Texas A&M University.

Klausen, C. & Russell, C. (2004), 'Relationships among some locally conservative discretization methods which handle discontinuous coefficients', *Comput. Geosci.* **8**, 341–377.

Klausen, R., Radu, F. & Eigestad, G. (2008), 'Convergence of MPFA on triangulations and for Richards' equation', *Int. J. Numer. Methods Fluids* pp. 1–25.

Klausen, R. & Winther, R. (2006*a*), 'Convergence of multipoint flux approximations on quadrilateral grids', *Numer. Meth. Part. D. E.* **22**, 1438–1454.

Klausen, R. & Winther, R. (2006*b*), 'Robust convergence of multi point flux approximation on rough grids', *Numer. Math.* **104**, 317–337.

Le Maître, O. & Knio, O. (2010), *Spectral Methods for Uncertainty Quantification With Applications to Computational Fluid Dynamics*, Springer-Verlag.

Loève, M. (1977), *Probability Theory*, 4th edn, Springer.

Ltd., E. (1996), Cardiff Bay Barrage groundwater characterisation, Technical report, Entec UK Ltd. and Hyder Consulting, Shrewsbury, UK.

Lu, Z. & Zhang, D. (2007), 'Stochastic simulations for flow in nonstationary randomly heterogeneous porous media using a KL-based moment-equation approach', *SIAM Multiscale Model. and Simul.* **6**(1), 228–245.

MATLAB (1997), Version 7.4.0, user's guide, Technical report, The Mathworks Inc, Prentice Hall.

Matthies, H. & Keese, A. (2005), 'Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations', *Comput. Methods Appl. Mech. Engrg.* **194**, 1295–1331.

McDonald, M. & Harbaugh, A. (1988), A modular three-dimensional finite differences ground-water flow model, Technical Report 83-875, U.S. Geological Survey, Reston, Virginia.

Mosé, R., Siegel, P., Ackerer, P. & Chavent, G. (1994), 'Application of the mixed hybrid finite element approximation in a groundwater flow model: Luxury or necessity?', *Water Resour. Res.* **30**(11), 3001–3012.

Najm, H. (2009), 'Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics', *Annual Review of Fluid Mechanics* **41**, 35–52.

Nedelec, J. (1980), 'Mixed finite elements in $\mathbb{R}^3$', *Numer. Math.* **35**, 315–341.

Nobile, F., Tempone, R. & Webster, C. (2008), 'A sparse grid stochastic collocation method for partial differential equations with random input data', *SIAM J. Numer. Anal.* **46**(5), 2309–2345.

Paige, C. & Saunders, M. (1975), 'Solution of sparse indefinite systems of linear equations', *SIAM J. Numer. Anal.* **12**, 617–629.

Pellissetti, M. F. & Ghanem, R. (2000), 'Iterative solution of systems of linear equations arising in the context of stochastic finite elements', *Adv. Eng. Softw.* **313**, 607–616.

Powell, C. E. (2003), Optimal Preconditioning for Mixed Finite Element Formulation of Second-Order Elliptic Problems, PhD thesis, The University of Manchester.

Powell, C. E. (2005), 'Parameter-free H(div) preconditioning for mixed finite element formulation of diffusion problems', *IMA J. Num. Anal.* **25**(4), 783–796.

Powell, C. & Elman, C. (2009), 'Block-diagonal preconditioning for spectral stochastic finite-element systems', *IMA J. Numer. Anal.* **29**(2), 350–375.

Powell, C. & Silvester, D. (2003), 'Optimal preconditioning for Raviart-Thomas mixed formulation of second-order elliptic problems', *SIAM J. Matrix Anal. A.* **25**(3), 718–738.

Powell, C. & Ullmann, E. (2010), 'Preconditioning stochastic Galerkin saddle point systems', *SIAM J. Matrix Anal. A.* **31**, 2813–2840.

Radu, A., Pop, I. & P., K. (2004), 'Order of convergence estimates for an Euler implicit, mixed finite element discretization of Richards' equation', *SIAM J. Numer. Anal.* **42**, 1452–1478.

Raviart, P. & Thomas, J. (1977), A mixed finite element method for second-order elliptic problems. Mathematical Aspects of the Finite Element Method, *in* 'Lect. Notes in Math.', Vol. 606, Springer-Verlag, pp. 292–315.

Roberts, J. & Thomas, J. (1991), Mixed and hybrid methods, *in* P. Ciarlet & J. Lions, eds, 'Handbook of Numerical Analysis', Elsevier Science Publishers, pp. 523–639.

Rossell, E., Boonen, T. & Vandewalle, S. (2008), 'Algebraic multigrid for stationary and time-dependent partial differential equations with stochastic coefficients', *Numer. Linear Algebra Appl.* **15**, 141–163.

Rossell, E. & Vandewalle, S. (2010), 'Iterative solvers for the stochastic finite element method', *SIAM J. Sci. Comp.* **32**(1), 372–397.

Rubin, Y. (2003), *Applied Stochastic Hydrogeology*, Oxford University Press.

Rusten, T., Vassilevski, P. & Wither, R. (1996), 'Interior preconditioners for mixed finite element approximations of elliptic problems', *Math. Comp.* **65**(214), 447–466.

Rusten, T. & Wither, R. (1992), 'A preconditioned iterative method for saddlepoint problems', *SIAM J. Matrix Anal. A.* **13**(3), 887–904.

Rusten, T. & Wither, R. (1993), 'Substructure preconditioners for elliptic saddle point problems', *Math. Comp.* **60**(201), 23–48.

Saad, Y. (2003), *Itarative Methods for Sparse Linear Systems*, SIAM.

Scheichl, R. (2000), Iterative Solution of Saddle Point Problems Using Divergence-free Finite Elements with Applications to Groundwater Flow, PhD thesis, University of Bath.

Scheichl, R. (2001), 'Decoupling three-dimensional mixed problems using divergence-free finite elements', *SIAM J. Sci. Comput.* **23**(5), 1752–1776.

Shen, J. (1994), Mixed finite element methods on distorted rectangular grids, Technical report, Institute for Scientific Computation, Texas A&M University.

Silvester, D. & Powell, C. (2007), PIFISS potential (incompressible) flow & iterative solution software guide, Technical report, University of Manchester.

Srivastava, R. & Brusseau, M. (1995), 'Darcy velocity computations in the finite element method for multidimensional randomly heterogeneous porous media', *Adv. Water Resour.* **18**(4), 191–201.

Stafanou, G. (2009), 'The stochastic finite element method: Past, present and future', *Applied Mechanics and Engineering* **198**, 1031–1051.

Stanley, P. (1995), 'The potential environmental risk to developments adjacent to, and the remediation of, the former Ferry Road municipal waste disposal site', Unpublished MSc. Thesis. School of Earth Sciences, Cardiff University.

Sudret, B. & Der Kiureghian, A. (2000), Stochastic finite element methods and reliability, a state-of-the-art-report, Technical Report UCB/SEMM-2000/08, University of California.

Sutton, J., Coulton, R. & Williams, B. (2004), 'Hydrogeology and groundwater control, Cardiff Bay Barrage', Urban Geology of Wales I, Geological series no.23. National Museum of Wales.

Tonkin, M. & Doherty, J. (2009), 'Calibration-constrained Monte Carlo analysis of highly parameterized models using subspace techniques', *Water Resour. Res.* **45**(W00B10), doi:10.1029/2007WR006678.

Ullmann, E. (2008), Solution Strategies for Stochastic Finite Element Discretizations, PhD thesis, Technische Universität Bergakademie Freiberg.

Van Wonderon, J. & Wilson, C. (2006), 5-yearly review of the Environment Agency's groundwater models, Technical Report 223060/01/C, Environment Agency of Englan and Wales.

Vassilevski, P. & Lazarov, R. (1996), 'Preconditioning mixed finite element saddle-point elliptic problems', *Numer. Linear Algebra Appl.* **3**(1), 1–20.

Vohralik, M. (2006), 'Equivalence between lowest-order mixed finite element and multi-point finite volume methods on simplicial meshes', *ESAIM: M2AN* **40**, 367–391.

Wheeler, M. & Yotov, I. (2006), 'A multipoint flux mixed finite element method', *SIAM J. Numer. Anal.* **44**, 2082–2106.

Williams, B. (2008), 'Cardiff Bay Barrage: management of groundwater issues', *Proceedings of the ICE - Water Management* **161**, 313–321.

Xiu, D. & Karniadakis, G. (2002*a*), 'Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos', *Comput. Methods Appl. Mech. Engrg.* **191**, 4927–4948.

Xiu, D. & Karniadakis, G. (2002*b*), 'The Wiener-Askey polynomial chaos for stochastic differential equations', *SIAM J. Sci. Comput.* **24**(2), 619–644.

Younes, A., Ackerer, P., Ahmed, S. & Bouhlila, R. (2010), 'A technique for improving the accuracy of quadrangular mixed finite elements for Darcy's flow on heterogeneous domains', *Computers & Fluids* **39**, 189–196.

Younes, A., Ackerer, P. & Chavent, G. (2004), 'From mixed finite elements to finite volumes for elliptic PDE in 2 and 3 dimensions', *Int. J. Numer. Meth. Eng.* **59**, 365–388.

Younes, A., Ackerer, P., Mosé, R. & Chavent, G. (1999), 'A new formulation of the mixed finite element method for solving elliptic and parabolic PDE with triangular elements', *J. Comput. Phys.* **149**, 148–167.

Younes, A. & Fontaine, V. (2008*a*), 'Efficiency of mixed hybrid finite element and multipoint flux approximation methods on quadrangular grids and highly anisotropic media', *Int. J. Numer. Meth. Eng.* **76**(3), 314–336.

Younes, A. & Fontaine, V. (2008*b*), 'Hybrid and multi-point formulations of the lowest-order mixed methods for Darcy's flow on triangles', *Int. J. Numer. Meth. Fl.* **58**(9), 1041–1062.

Zander, E. (2010), 'Stochastic Galerkin library, *sglib*'. https://github.com/ezander/sglib.

# Appendix A

# Set-up Time for Test Problems and Preconditioner $\mathcal{P}_{bdiag}$

## A.1 Set-up time for test problems and $\mathcal{P}_{bdiag}$ preconditioner - Linear case

Table A.1: Problem and $\mathcal{P}_{bdiag}$ set-up times (sec.) - Test Problem 1

| | $h$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *cholinc* | $\frac{1}{32}$ | $1.21 + 0.01$ | $0.15 + 0.01$ | $0.15 + 0.01$ |
| | $\frac{1}{64}$ | $0.23 + 0.06$ | $0.23 + 0.06$ | $0.23 + 0.06$ |
| | $\frac{1}{128}$ | $0.66 + 0.41$ | $0.66 + 0.44$ | $0.66 + 0.44$ |
| *AMG* | $\frac{1}{32}$ | $0.88 + 0.31$ | $0.15 + 0.27$ | $0.15 + 0.27$ |
| | $\frac{1}{64}$ | $0.23 + 1.08$ | $0.22 + 1.06$ | $0.22 + 1.08$ |
| | $\frac{1}{128}$ | $0.66 + 7.31$ | $0.66 + 7.61$ | $0.66 + 7.31$ |
| $d = 6$ | | | | |
| *cholinc* | $\frac{1}{32}$ | $0.16 + 0.01$ | $0.16 + 0.01$ | $0.17 + 0.01$ |
| | $\frac{1}{64}$ | $0.26 + 0.06$ | $0.26 + 0.06$ | $0.26 + 0.06$ |
| | $\frac{1}{128}$ | $0.83 + 0.43$ | $0.85 + 0.44$ | $0.87 + 0.44$ |
| *AMG* | $\frac{1}{32}$ | $0.17 + 0.27$ | $0.16 + 0.27$ | $0.16 + 0.27$ |
| | $\frac{1}{64}$ | $0.26 + 1.06$ | $0.26 + 1.08$ | $0.27 + 1.05$ |
| | $\frac{1}{128}$ | $0.86 + 7.59$ | $0.85 + 7.43$ | $0.86 + 7.34$ |

Table A.2: Problem and $\mathcal{P}_{bdiag}$ set-up times (sec.) - Test Problem 2

| | $\sigma$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.3 | $1.42 + 0.01$ | $0.14 + 0.01$ | $0.15 + 0.01$ |
| *cholinc* | 0.5 | $0.15 + 0.01$ | $0.15 + 0.01$ | $0.15 + 0.01$ |
| | 0.7 | $0.14 + 0.01$ | $0.14 + 0.01$ | $0.15 + 0.01$ |
| | 0.3 | $0.58 + 0.34$ | $0.14 + 0.28$ | $0.15 + 0.28$ |
| *AMG* | 0.5 | $0.15 + 0.28$ | $0.14 + 0.28$ | $0.15 + 0.28$ |
| | 0.7 | $0.14 + 0.28$ | $0.15 + 0.28$ | $0.15 + 0.28$ |
| $d = 6$ | | | | |
| | 0.3 | $0.16 + 0.01$ | $0.16 + 0.01$ | $0.17 + 0.01$ |
| *cholinc* | 0.5 | $0.15 + 0.01$ | $0.15 + 0.01$ | $0.17 + 0.01$ |
| | 0.7 | $0.15 + 0.01$ | $0.15 + 0.01$ | $0.17 + 0.01$ |
| | 0.3 | $0.15 + 0.28$ | $0.15 + 0.28$ | $0.17 + 0.28$ |
| *AMG* | 0.5 | $0.15 + 0.28$ | $0.15 + 0.28$ | $0.17 + 0.28$ |
| | 0.7 | $0.15 + 0.28$ | $0.16 + 0.28$ | $0.17 + 0.28$ |

Table A.3: Problem and $\mathcal{P}_{bdiag}$ set-up times (sec.) - Test Problem 3

| | $\delta = \frac{\sigma}{\mu}$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.3 | $0.17 + 0.01$ | $0.14 + 0.01$ | $0.15 + 0.01$ |
| *cholinc* | 0.5 | $0.14 + 0.01$ | $0.14 + 0.01$ | $0.14 + 0.01$ |
| | 0.7 | $0.15 + 0.01$ | $0.14 + 0.01$ | $0.15 + 0.01$ |
| | 0.7,0.5,0.6,0.7 | $0.15 + 0.01$ | $0.14 + 0.01$ | $0.14 + 0.01$ |
| | 0.3 | $0.17 + 0.32$ | $0.14 + 0.32$ | $0.14 + 0.32$ |
| *AMG* | 0.5 | $0.15 + 0.32$ | $0.14 + 0.32$ | $0.14 + 0.32$ |
| | 0.7 | $0.14 + 0.32$ | $0.14 + 0.32$ | $0.14 + 0.32$ |
| | 0.7,0.5,0.6,0.7 | $0.14 + 0.32$ | $0.14 + 0.32$ | $0.14 + 0.32$ |
| $d = 6$ | | | | |
| | 0.5 | $0.16 + 0.01$ | $0.16 + 0.01$ | $0.17 + 0.01$ |
| *cholinc* | 0.7 | $0.16 + 0.01$ | $0.16 + 0.01$ | $0.17 + 0.01$ |
| | 1.0 | $0.16 + 0.01$ | $0.16 + 0.01$ | $0.17 + 0.01$ |
| | 0.7,0.5,0.6,0.7 | $0.16 + 0.01$ | $0.16 + 0.01$ | $0.17 + 0.01$ |
| | 0.3 | $0.16 + 0.32$ | $0.16 + 0.32$ | $0.2 + 0.32$ |
| *AMG* | 0.5 | $0.16 + 0.32$ | $0.16 + 0.32$ | $0.17 + 0.32$ |
| | 0.7 | $0.16 + 0.32$ | $0.16 + 0.32$ | $0.17 + 0.32$ |
| | 0.7,0.5,0.6,0.7 | $0.16 + 0.32$ | $0.16 + 0.32$ | $0.17 + 0.32$ |

## A.2 Set-up time for test problems and $\mathcal{P}_{bdiag}$ preconditioner - Nonlinear case

Table A.4: Problem and $\mathcal{P}_{bdiag}$ $(AMG)$ set-up times (sec.) - Test Problem 1

| | $h$ | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 1.59 | 1.66 | 5.76 |
| | $\frac{1}{64}$ | 1.47 | 4.6 | 12.85 |
| | $\frac{1}{128}$ | 7.15 | 21.57 | 53.75 |
| *AMG* | $\frac{1}{32}$ | $0.54 + 0.27$ | $1.8 + 0.31$ | $5.82 + 0.27$ |
| | $\frac{1}{64}$ | $1.47 + 1.07$ | $4.58 + 1.07$ | $12.91 + 1.08$ |
| | $\frac{1}{128}$ | $7.15 + 7.67$ | $21.67 + 7.48$ | $53.99 + 7.84$ |
| $d = 6$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 2.01 | 22.92 | 221.02 |
| | $\frac{1}{64}$ | 4.9 | 36.54 | 262.61 |
| | $\frac{1}{128}$ | 21.45 | 109.71 | 625.23 |
| *AMG* | $\frac{1}{32}$ | $1.97 + 0.27$ | $23.07 + 0.27$ | $217.67 + 0.28$ |
| | $\frac{1}{64}$ | $4.78 + 1.08$ | $36.73 + 1.09$ | $257.69 + 1.16$ |
| | $\frac{1}{128}$ | $21.27 + 7.84$ | $110.71 + 7.86$ | $698.77 + 14.2$ |

Table A.5: Problem and $\mathcal{P}_{bdiag}$ $(AMG)$ set-up times (sec.) - Test Problem 2

| | $h$ | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *UMFPACK* | 0.3 | 0.52 | 1.61 | 5.74 |
| | 0.5 | 0.48 | 1.62 | 5.83 |
| | 0.7 | 0.48 | 1.62 | 5.76 |
| | 0.9 | 0.49 | 1.61 | 5.81 |
| *AMG* | 0.3 | $1.84 + 0.36$ | $1.61 + 0.28$ | $5.68 + 0.28$ |
| | 0.5 | $0.48 + 0.28$ | $1.59 + 0.28$ | $5.66 + 0.28$ |
| | 0.7 | $0.48 + 0.28$ | $1.59 + 0.28$ | $5.75 + 0.28$ |
| | 0.9 | $0.48 + 0.28$ | $1.59 + 0.28$ | $5.59 + 0.28$ |
| $d = 6$ | | | | |
| *UMFPACK* | 0.3 | 1.99 | 22.86 | 211.46 |
| | 0.5 | 1.91 | 22.43 | 212.54 |
| | 0.7 | 1.93 | 22.79 | 215.94 |
| | 0.9 | 1.93 | 22.56 | 223.39 |
| *AMG* | 0.3 | $1.89 + 0.28$ | $22.78 + 0.28$ | $216.79 + 0.29$ |
| | 0.5 | $1.88 + 0.28$ | $22.53 + 0.28$ | $212.85 + 0.29$ |
| | 0.7 | $1.88 + 0.28$ | $22.21 + 0.28$ | $221.46 + 0.28$ |
| | 0.9 | $1.87 + 0.28$ | $22.54 + 0.28$ | $220.42 + 0.29$ |

Table A.6: Problem and $\mathcal{P}_{bdiag}$ $(AMG)$ set-up times (sec.) - Test Problem 3

| | $\delta$ | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.5 | 0.58 | 1.83 | 6.26 |
| *UMFPACK* | 0.7 | 0.56 | 1.81 | 6.2 |
| | 1.0 | 0.56 | 1.83 | 6.19 |
| | 1.0,0.7,0.5,1.0 | 0.55 | 1.82 | 6.2 |
| | 0.5 | $0.58 + 0.3$ | $1.81 + 0.29$ | $6.3 + 0.29$ |
| *AMG* | 0.7 | $0.54 + 0.3$ | $1.81 + 0.3$ | $6.2 + 0.3$ |
| | 1.0 | $0.54 + 0.31$ | $1.81 + 0.31$ | $6.23 + 0.31$ |
| | 1.0,0.7,0.5,1.0 | $0.54 + 0.31$ | $1.79 + 0.31$ | $6.27 + 0.31$ |
| $d = 6$ | | | | |
| | 0.5 | 2.13 | 23.49 | 222.3 |
| *UMFPACK* | 0.7 | 2.13 | 23.66 | 220.6 |
| | 1.0 | 2.13 | 23.78 | 224.36 |
| | 1.0,0.7,0.5,1.0 | 2.12 | 24.31 | 219.31 |
| | 0.5 | $2.11 + 0.29$ | $23.44 + 0.3$ | $223.16 + 0.3$ |
| *AMG* | 0.7 | $2.12 + 0.3$ | $23.88 + 0.3$ | $222.08 + 0.31$ |
| | 1.0 | $2.13 + 0.31$ | $23.42 + 0.31$ | $222.11 + 0.32$ |
| | 1.0,0.7,0.5,1.0 | $2.13 + 0.31$ | $23.42 + 0.31$ | $226.08 + 0.32$ |

# Appendix B

# Simulation Results for $\mathcal{P}_{mean}$ Preconditioner

## B.1 Simulations for $\mathcal{P}_{mean}$ preconditioner - Linear case

Table B.1: CG iterations and solution timings for $\mathcal{P}_{mean}$ - Test Problem 1

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| *cholinc* | $\frac{1}{32}$ | 11 | 0.18 | 17 | 0.49 | 29 | 1.78 |
| | $\frac{1}{64}$ | 13 | 0.59 | 21 | 2.27 | 37 | 8.61 |
| | $\frac{1}{128}$ | 21 | 9.46 | 35 | 38.98 | 64 | 157.69 |
| *AMG* | $\frac{1}{32}$ | 12 | 0.58 | 19 | 1.39 | 33 | 4.9 |
| | $\frac{1}{64}$ | 12 | 0.78 | 19 | 2.97 | 35 | 11.44 |
| | $\frac{1}{128}$ | 13 | 3.49 | 20 | 13.23 | 35 | 48.67 |
| $d = 6$ | | | | | | | |
| *cholinc* | $\frac{1}{32}$ | 11 | 0.27 | 17 | 1.26 | 30 | 6.16 |
| | $\frac{1}{64}$ | 13 | 1.12 | 21 | 5.9 | 37 | 28.08 |
| | $\frac{1}{128}$ | 21 | 18.56 | 35 | 95.17 | 64 | 448.33 |
| *AMG* | $\frac{1}{32}$ | 12 | 0.71 | 19 | 3.45 | 33 | 15.41 |
| | $\frac{1}{64}$ | 12 | 1.48 | 20 | 7.95 | 35 | 36.46 |
| | $\frac{1}{128}$ | 13 | 6.68 | 20 | 33.51 | 37 | 163.68 |

Table B.2: CG iterations and solution timings for $\mathcal{P}_{mean}$ - Test Problem 2

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| *cholinc* | 0.3 | 15 | 0.62 | 22 | 0.66 | 29 | 1.85 |
| | 0.5 | 18 | 0.22 | 26 | 0.78 | 38 | 2.43 |
| | 0.7 | 21 | 0.26 | 34 | 1.02 | 54 | 3.44 |
| *AMG* | 0.3 | 18 | 0.66 | 28 | 2.26 | 40 | 6.57 |
| | 0.5 | 20 | 0.69 | 33 | 2.66 | 49 | 8.05 |
| | 0.7 | 24 | 0.82 | 38 | 3.11 | 65 | 10.67 |
| $d = 6$ | | | | | | | |
| *cholinc* | 0.3 | 15 | 0.35 | 22 | 1.71 | 29 | 6.17 |
| | 0.5 | 18 | 0.42 | 28 | 2.15 | 42 | 8.95 |
| | 0.7 | 22 | 0.51 | 38 | 2.91 | 72 | 15.37 |
| *AMG* | 0.3 | 18 | 1.18 | 28 | 5.57 | 40 | 20.68 |
| | 0.5 | 21 | 1.34 | 33 | 6.53 | 51 | 26.33 |
| | 0.7 | 24 | 1.52 | 43 | 8.48 | 82 | 42.36 |

Table B.3: CG iterations and solution timings for $\mathcal{P}_{mean}$ - Test Problem 3

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| *cholinc* | 0.3 | 20 | 0.23 | 33 | 0.63 | 53 | 2.03 |
| | 0.5 | 23 | 0.22 | 39 | 0.8 | 61 | 2.27 |
| | 0.7 | 25 | 0.23 | 44 | 0.94 | 74 | 3.11 |
| | 0.7,0.5,0.6,0.7 | 24 | 0.23 | 42 | 0.84 | 69 | 2.75 |
| *AMG* | 0.3 | 18 | 0.73 | 28 | 2.54 | 36 | 6.6 |
| | 0.5 | 20 | 0.77 | 31 | 2.79 | 45 | 8.26 |
| | 0.7 | 23 | 0.88 | 36 | 3.23 | 55 | 10.1 |
| | 0.7,0.5,0.6,0.7 | 21 | 0.81 | 34 | 3.06 | 50 | 9.16 |
| $d = 6$ | | | | | | | |
| *cholinc* | 0.3 | 19 | 0.35 | 33 | 1.84 | 55 | 6.69 |
| | 0.5 | 21 | 0.42 | 37 | 2.06 | 62 | 8.55 |
| | 0.7 | 25 | 0.48 | 46 | 2.64 | 83 | 12.63 |
| | 0.7,0.5,0.6,0.7 | 25 | 0.44 | 43 | 2.42 | 75 | 11.41 |
| *AMG* | 0.3 | 18 | 1.29 | 29 | 6.41 | 40 | 22.91 |
| | 0.5 | 20 | 1.42 | 32 | 7.05 | 48 | 27.47 |
| | 0.7 | 23 | 1.67 | 40 | 8.81 | 68 | 38.89 |
| | 0.7,0.5,0.6,0.7 | 21 | 1.49 | 36 | 7.94 | 60 | 34.39 |

## B.2 Simulations for $\mathcal{P}_{mean}$ preconditioner - Nonlinear Case

Table B.4: CG iterations and solution timings for $\mathcal{P}_{mean}$ - Test Problem 1

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 11 | 1.06 | 17 | 5.56 | 31 | 50.88 |
| | $\frac{1}{64}$ | 11 | 2.8 | 17 | 20.85 | 31 | 167.48 |
| | $\frac{1}{128}$ | 11 | 15.47 | 17 | 89.8 | 31 | 671.73 |
| *AMG* | $\frac{1}{32}$ | 12 | 1.01 | 19 | 6.3 | 35 | 57.54 |
| | $\frac{1}{64}$ | 12 | 2.23 | 20 | 21.41 | 37 | 186.8 |
| | $\frac{1}{128}$ | 13 | 9.28 | 21 | 86.35 | 38 | 729.57 |
| $d = 6$ | | | | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 11 | 1.94 | 17 | 30.41 | 31 | 431.97 |
| | $\frac{1}{64}$ | 11 | 7.54 | 17 | 100.33 | 31 | $1,349.11$ |
| | $\frac{1}{128}$ | 11 | 33.79 | 17 | 405.59 | 31 | $5,208.41$ |
| *AMG* | $\frac{1}{32}$ | 12 | 2.09 | 19 | 34.21 | 37 | 515.77 |
| | $\frac{1}{64}$ | 13 | 7.23 | 21 | 115.74 | 37 | $1,593.79$ |
| | $\frac{1}{128}$ | 13 | 27.49 | 21 | 441.14 | 39 | $6,323.83$ |

Table B.5: CG iterations and solution timings for $\mathcal{P}_{mean}$ - Test Problem 2

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 15 | 1.15 | 29 | 9.8 | 58 | 96.53 |
| *UMFPACK* | 0.5 | 22 | 1.31 | 46 | 15.51 | 102 | 170.28 |
| | 0.7 | 30 | 1.78 | 69 | 23.17 | 171 | 285.7 |
| | 0.9 | 39 | 2.3 | 103 | 34.7 | 278 | 463.81 |
| | 0.3 | 17 | 1.33 | 31 | 10.82 | 62 | 105.46 |
| *AMG* | 0.5 | 23 | 1.46 | 48 | 16.65 | 107 | 180.8 |
| | 0.7 | 31 | 1.95 | 71 | 24.61 | 177 | 299.15 |
| | 0.9 | 41 | 2.58 | 106 | 36.59 | 284 | 480.69 |
| $d = 6$ | | | | | | | |
| | 0.3 | 15 | 2.62 | 29 | 53.24 | 60 | 837.37 |
| *UMFPACK* | 0.5 | 22 | 3.84 | 47 | 85.43 | 107 | 1,486.98 |
| | 0.7 | 30 | 5.27 | 72 | 129.58 | 180 | 2,498.76 |
| | 0.9 | 40 | 6.96 | 107 | 193.75 | 294 | 4,102.17 |
| | 0.3 | 17 | 3.16 | 32 | 58.68 | 65 | 913.56 |
| *AMG* | 0.5 | 23 | 4.2 | 49 | 90.11 | 112 | 1,577.59 |
| | 0.7 | 32 | 5.83 | 75 | 138.23 | 186 | 2,627.54 |
| | 0.9 | 42 | 7.65 | 110 | 202.08 | 301 | 4,252.37 |

Table B.6: CG iterations and solution timings for $\mathcal{P}_{mean}$ - Test Problem 3

| | | $p_u = 2$ | | $p_u = 3$ | | $p_u = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.5 | 20 | 1.24 | 40 | 13.5 | 86 | 144.3 |
| *UMFPACK* | 0.7 | 26 | 1.54 | 59 | 19.92 | 138 | 231.21 |
| | 1.0 | 41 | 2.44 | 100 | 33.79 | 259 | 434.85 |
| | 1.0,0.7,0.5,1.0 | 40 | 2.38 | 100 | 33.9 | 259 | 434.39 |
| | 0.5 | 21 | 1.77 | 42 | 14.5 | 91 | 153.19 |
| *AMG* | 0.7 | 28 | 1.94 | 60 | 20.75 | 142 | 238.62 |
| | 1.0 | 42 | 2.76 | 101 | 34.89 | 266 | 448.52 |
| | 1.0,0.7,0.5,1.0 | 42 | 2.67 | 101 | 34.88 | 265 | 446.67 |
| $d = 6$ | | | | | | | |
| | 0.5 | 20 | 3.53 | 42 | 76.39 | 90 | $1,263.25$ |
| *UMFPACK* | 0.7 | 28 | 4.91 | 61 | 111.51 | 145 | $2,040.56$ |
| | 1.0 | 42 | 7.38 | 105 | 190.57 | 280 | $3,925.73$ |
| | 1.0,0.7,0.5,1.0 | 42 | 7.37 | 105 | 190.74 | 280 | $3,918.66$ |
| | 0.5 | 22 | 4.01 | 43 | 78.62 | 96 | $1,351.2$ |
| *AMG* | 0.7 | 29 | 5.26 | 63 | 115.41 | 152 | $2,144.95$ |
| | 1.0 | 44 | 7.97 | 107 | 196.11 | 287 | $4,016.72$ |
| | 1.0,0.7,0.5,1.0 | 44 | 8.01 | 107 | 195.57 | 289 | $4,029.83$ |

# Appendix C

# Set-up Times for Test Problems and Preconditioner $\mathcal{P}_{bSGS}$

## C.1 Set-up time for test problems and $\mathcal{P}_{bSGS}$ preconditioner - Linear case

Table C.1: Problem and $\mathcal{P}_{bSGS}$ set-up times (sec.) - Test Problem 1

| | $h$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 1.6 | 1.64 | 5.8 |
| | $\frac{1}{64}$ | 1.48 | 4.56 | 12.94 |
| | $\frac{1}{128}$ | 7.05 | 21.61 | 54.04 |
| *AMG* | $\frac{1}{32}$ | $1.85 + 0.34$ | $1.64 + 0.27$ | $5.89 + 0.27$ |
| | $\frac{1}{64}$ | $1.47 + 1.07$ | $4.59 + 1.06$ | $12.92 + 1.05$ |
| | $\frac{1}{128}$ | $7.14 + 7.21$ | $21.57 + 7.12$ | $53.96 + 8.43$ |
| $d = 6$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 1.99 | 23 | 220.26 |
| | $\frac{1}{64}$ | 4.88 | 36.08 | 261.8 |
| | $\frac{1}{128}$ | 21.31 | 110.29 | 673.95 |
| *AMG* | $\frac{1}{32}$ | $1.98 + 0.27$ | $22.63 + 0.27$ | $215.64 + 0.28$ |
| | $\frac{1}{64}$ | $4.85 + 1.1$ | $36.15 + 1.12$ | $263.51 + 1.08$ |
| | $\frac{1}{128}$ | $21.42 + 7.32$ | $110.13 + 8.37$ | $606.56 + 12.32$ |

Table C.2: Problem and $\mathcal{P}_{bSGS}$ set-up times (sec.) - Test Problem 2

| | $\sigma$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.3 | 0.52 | 1.59 | 5.77 |
| *UMFPACK* | 0.5 | 0.48 | 1.59 | 5.81 |
| | 0.7 | 0.48 | 1.58 | 5.77 |
| | 0.9 | 0.48 | 1.61 | 5.7 |
| | 0.3 | 1.6 + 0.29 | 1.6 + 0.28 | 5.64 + 0.28 |
| *AMG* | 0.5 | 0.48 + 0.28 | 1.58 + 0.28 | 5.68 + 0.28 |
| | 0.7 | 0.48 + 0.28 | 1.59 + 0.28 | 5.68 + 0.28 |
| | 0.9 | 0.48 + 0.28 | 1.59 + 0.28 | 5.64 + 0.28 |
| $d = 6$ | | | | |
| | 0.3 | 1.93 | 22.79 | 218.77 |
| *UMFPACK* | 0.5 | 1.9 | 22.73 | 216.69 |
| | 0.7 | 1.89 | 22.9 | 214.72 |
| | 0.9 | 1.89 | 22.84 | 214.83 |
| | 0.3 | 1.9 + 0.28 | 22.77 + 0.28 | 214.76 + 0.29 |
| *AMG* | 0.5 | 1.9 + 0.28 | 22.59 + 0.28 | 214.26 + 0.28 |
| | 0.7 | 1.9 + 0.28 | 22.96 + 0.28 | 221.55 + 0.29 |
| | 0.9 | 1.92 + 0.28 | 22.69 + 0.28 | 218.06 + 0.29 |

Table C.3: Problem and $\mathcal{P}_{bSGS}$ set-up times (sec.) - Test Problem 3

| | $\delta = \frac{\sigma}{\mu}$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.5 | 0.63 | 1.82 | 6.31 |
| *UMFPACK* | 0.7 | 0.55 | 1.81 | 6.19 |
| | 1.0 | 0.54 | 1.83 | 6.27 |
| | 1.0,0.7,0.5,1.0 | 0.54 | 1.82 | 6.27 |
| | 0.5 | 1.86 + 0.37 | 1.81 + 0.29 | 6.36 + 0.3 |
| *AMG* | 0.7 | 0.56 + 0.3 | 1.79 + 0.3 | 6.27 + 0.3 |
| | 1.0 | 0.54 + 0.31 | 1.81 + 0.31 | 6.38 + 0.31 |
| | 1.0,0.7,0.5,1.0 | 0.54 + 0.31 | 1.8 + 0.31 | 6.24 + 0.31 |
| $d = 6$ | | | | |
| | 0.5 | 2.14 | 23.61 | 215.04 |
| *UMFPACK* | 0.7 | 2.15 | 23.47 | 223.14 |
| | 1.0 | 2.14 | 24.21 | 220.55 |
| | 1.0,0.7,0.5,1.0 | 2.15 | 23.37 | 218.9 |
| | 0.5 | 2.11 + 0.29 | 23.56 + 0.3 | 219.35 + 0.3 |
| *AMG* | 0.7 | 2.11 + 0.3 | 23.99 + 0.3 | 220.22 + 0.31 |
| | 1.0 | 2.12 + 0.31 | 24.04 + 0.31 | 222.92 + 0.32 |
| | 1.0,0.7,0.5,1.0 | 2.1 + 0.31 | 23.83 + 0.32 | 223.62 + 0.32 |

## C.2 Set-up time for test problems and $\mathcal{P}_{bSGS}$ preconditioner - Nonlinear case

Table C.4: Problem and $\mathcal{P}_{bSGS}$ set-up times (sec.) - Test Problem 1

| | $h$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | $\frac{1}{32}$ | 1.25 | 1.49 | 5.55 |
| *UMFPACK* | $\frac{1}{64}$ | 1.32 | 4.44 | 12.72 |
| | $\frac{1}{128}$ | 6.98 | 21.41 | 54 |
| | $\frac{1}{32}$ | $3.23 + 3.15$ | $1.67 + 6.48$ | $5.88 + 12.91$ |
| *AMG* | $\frac{1}{64}$ | $1.47 + 14.8$ | $4.52 + 34.19$ | $12.84 + 69.16$ |
| | $\frac{1}{128}$ | $7.16 + 114.43$ | $21.5 + 269.13$ | $54.13 + 590.05$ |
| $d = 6$ | | | | |
| | $\frac{1}{32}$ | 1.79 | 22.17 | 210.54 |
| *UMFPACK* | $\frac{1}{64}$ | 4.66 | 35.88 | 254.69 |
| | $\frac{1}{128}$ | 21.22 | 109.91 | 565.43 |
| | $\frac{1}{32}$ | $1.97 + 5.21$ | $22.92 + 15.56$ | $221.74 + 39.43$ |
| *AMG* | $\frac{1}{64}$ | $4.87 + 27.74$ | $35.88 + 86.98$ | $263.95 + 215.93$ |
| | $\frac{1}{128}$ | $21.38 + 246.93$ | $109.29 + 810.26$ | $620.38 + 2014.46$ |

Table C.5: Problem and $\mathcal{P}_{bSGS}$ set-up times (sec.) - Test Problem 2

| | $h$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.3 | 1.86 | 1.47 | 5.7 |
| *UMFPACK* | 0.5 | 0.34 | 1.47 | 5.73 |
| | 0.7 | 0.33 | 1.47 | 5.73 |
| | 0.9 | 0.33 | 1.47 | 5.73 |
| | 0.3 | $1.66 + 3.06$ | $1.59 + 6.89$ | $5.76 + 13.77$ |
| *AMG* | 0.5 | $0.48 + 2.98$ | $1.58 + 6.88$ | $5.72 + 13.78$ |
| | 0.7 | $0.48 + 3$ | $1.58 + 6.81$ | $5.71 + 13.67$ |
| | 0.9 | $0.48 + 2.99$ | $1.58 + 6.91$ | $5.71 + 13.73$ |
| $d = 6$ | | | | |
| | 0.3 | 1.8 | 23.23 | 223.37 |
| *UMFPACK* | 0.5 | 1.79 | 22.99 | 219.91 |
| | 0.7 | 1.79 | 23.01 | 220.51 |
| | 0.9 | 1.79 | 22.98 | 220.33 |
| | 0.3 | $1.9 + 5.95$ | $22.69 + 16.53$ | $219.58 + 41.36$ |
| *AMG* | 0.5 | $1.87 + 5.51$ | $22.78 + 16.52$ | $215.15 + 41.68$ |
| | 0.7 | $1.9 + 5.52$ | $22.9 + 16.47$ | $219.37 + 41.87$ |
| | 0.9 | $1.93 + 5.53$ | $22.27 + 16.6$ | $215.84 + 41.88$ |

Table C.6: Problem and $\mathcal{P}_{bSGS}$ set-up times (sec.) - Test Problem 3

| | $h$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.5 | 0.58 | 1.68 | 6.09 |
| *UMFPACK* | 0.7 | 0.41 | 1.67 | 6.07 |
| | 1.0 | 0.41 | 1.67 | 6.07 |
| | 1.0,0.7,0.5,1.0 | 0.41 | 1.67 | 6.11 |
| | 0.5 | $1.52 + 3.21$ | $1.82 + 7.08$ | $6.23 + 14.2$ |
| *AMG* | 0.7 | $0.54 + 3.18$ | $1.8 + 7.31$ | $6.23 + 14.75$ |
| | 1.0 | $0.54 + 3.32$ | $1.81 + 7.64$ | $6.32 + 15.05$ |
| | 1.0,0.7,0.5,1.0 | $0.54 + 3.34$ | $1.79 + 7.68$ | $6.11 + 15.26$ |
| $d = 6$ | | | | |
| | 0.5 | 1.99 | 23.26 | 214.8 |
| *UMFPACK* | 0.7 | 1.99 | 23.24 | 217.64 |
| | 1.0 | 1.99 | 23.31 | 216.9 |
| | 1.0,0.7,0.5,1.0 | 1.97 | 23.32 | 218.9 |
| | 0.5 | $2.14 + 5.74$ | $23.76 + 17.14$ | $225.16 + 42.47$ |
| *AMG* | 0.7 | $2.14 + 5.93$ | $24.03 + 17.53$ | $222.89 + 44.73$ |
| | 1.0 | $2.13 + 6.23$ | $24.13 + 18.38$ | $221.56 + 46.59$ |
| | 1.0,0.7,0.5,1.0 | $2.11 + 6.22$ | $23.9 + 18.46$ | $227.19 + 46.53$ |

# Appendix D

# Numerical Simulations for Gauss-Siedel solvers

## D.1 Simulations for Gauss-Siedel solvers - Linear case

Table D.1: bSGS and bGS iterations and solution timings (*UMFPACK* case) - Test Problem 1

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $h$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| $bSGS$ | $\frac{1}{32}$ | 4 | 0.19 | 4 | 0.43 | 4 | 0.87 |
| | $\frac{1}{64}$ | 5 | 1.14 | 5 | 2.67 | 5 | 5.38 |
| | $\frac{1}{128}$ | 5 | 8.41 | 5 | 17.44 | 5 | 35.09 |
| $bGS$ | $\frac{1}{32}$ | 6 | 0.17 | 6 | 0.32 | 7 | 0.76 |
| | $\frac{1}{64}$ | 6 | 0.68 | 7 | 1.88 | 7 | 3.74 |
| | $\frac{1}{128}$ | 6 | 5.59 | 7 | 11.43 | 7 | 23.06 |
| $d = 6$ | | | | | | | |
| $bSGS$ | $\frac{1}{32}$ | 4 | 0.35 | 4 | 1.04 | 4 | 2.65 |
| | $\frac{1}{64}$ | 5 | 2.13 | 5 | 6.45 | 5 | 16.24 |
| | $\frac{1}{128}$ | 5 | 13.91 | 5 | 41.82 | 5 | 105.89 |
| $bGS$ | $\frac{1}{32}$ | 6 | 0.26 | 6 | 0.78 | 7 | 2.33 |
| | $\frac{1}{64}$ | 6 | 1.28 | 7 | 4.56 | 7 | 11.42 |
| | $\frac{1}{128}$ | 6 | 7.82 | 7 | 27.6 | 7 | 69.73 |

Table D.2: bSGS and bGS iterations and solution timings (*UMFPACK* case) - Test Problem 2

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\sigma$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 6 | 0.29 | 6 | 0.69 | 6 | 1.38 |
| *bSGS* | 0.5 | 8 | 0.39 | 9 | 1.02 | 10 | 2.3 |
| | 0.7 | 12 | 0.58 | 17 | 1.93 | 24 | 5.5 |
| | 0.3 | 8 | 0.2 | 8 | 0.47 | 9 | 1.04 |
| *bGS* | 0.5 | 11 | 0.28 | 13 | 0.76 | 15 | 1.77 |
| | 0.7 | 16 | 0.39 | 24 | 1.4 | 36 | 4.24 |
| $d = 6$ | | | | | | | |
| | 0.3 | 6 | 0.55 | 6 | 1.66 | 6 | 4.22 |
| *bSGS* | 0.5 | 9 | 0.81 | 10 | 2.76 | 12 | 8.44 |
| | 0.7 | 13 | 1.17 | 20 | 5.54 | 39 | 27.37 |
| | 0.3 | 8 | 0.38 | 8 | 1.14 | 9 | 3.25 |
| *bGS* | 0.5 | 11 | 0.51 | 14 | 1.98 | 17 | 6.17 |
| | 0.7 | 17 | 0.79 | 29 | 4.07 | 60 | 21.73 |

Table D.3: bSGS and bGS iterations and solution timings (*UMFPACK* case) - Test Problem 3

| | | $p = 2$ | | $p = 3$ | | $p = 4$ | |
|---|---|---|---|---|---|---|---|
| | $\delta = \frac{\sigma}{\mu}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ | $N_{it}$ | $t_{CPU}$ |
| | | | (sec.) | | (sec.) | | (sec.) |
| $d = 4$ | | | | | | | |
| | 0.3 | 6 | 0.33 | 6 | 0.69 | 6 | 1.38 |
| *bSGS* | 0.5 | 8 | 0.39 | 9 | 1.02 | 10 | 2.29 |
| | 0.7 | 11 | 0.53 | 14 | 1.61 | 18 | 4.13 |
| | 0.7,0.5,0.6,0.7 | 11 | 0.53 | 14 | 1.59 | 18 | 4.12 |
| | 0.3 | 7 | 0.17 | 8 | 0.46 | 8 | 0.93 |
| *bGS* | 0.5 | 10 | 0.24 | 12 | 0.69 | 14 | 1.62 |
| | 0.7 | 14 | 0.34 | 20 | 1.14 | 28 | 3.24 |
| | 0.7,0.5,0.6,0.7 | 14 | 0.34 | 20 | 1.14 | 27 | 3.12 |
| $d = 6$ | | | | | | | |
| | 0.3 | 6 | 0.55 | 6 | 1.65 | 6 | 4.21 |
| *bSGS* | 0.5 | 8 | 0.72 | 10 | 2.76 | 11 | 7.72 |
| | 0.7 | 12 | 1.08 | 18 | 4.96 | 30 | 21.02 |
| | 0.7,0.5,0.6,0.7 | 12 | 1.08 | 18 | 4.94 | 30 | 21.01 |
| | 0.3 | 8 | 0.37 | 8 | 1.11 | 9 | 3.19 |
| *bGS* | 0.5 | 11 | 0.5 | 13 | 1.81 | 16 | 5.69 |
| | 0.7 | 16 | 0.73 | 25 | 3.45 | 46 | 16.28 |
| | 0.7,0.5,0.6,0.7 | 16 | 0.73 | 25 | 3.46 | 46 | 16.25 |

# Appendix E

# Set-up Times for Test Problems and Preconditioner $\mathcal{P}_{Schur}$

## E.1 Set-up times for test problems and $\mathcal{P}_{Schur}$ preconditioner - Linear case

Table E.1: Problem and $\mathcal{P}_{Schur}$ set-up times (sec.) - Test Problem 1

| | $h$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 0.72 | 0.18 | 0.18 |
| | $\frac{1}{64}$ | 0.29 | 0.29 | 0.29 |
| | $\frac{1}{128}$ | 1.22 | 1.24 | 1.26 |
| *AMG* | $\frac{1}{32}$ | $1.02 + 0.57$ | $0.18 + 0.49$ | $0.18 + 0.49$ |
| | $\frac{1}{64}$ | $0.29 + 2.5$ | $0.29 + 2.51$ | $0.29 + 2.46$ |
| | $\frac{1}{128}$ | $1.23 + 22.03$ | $1.19 + 22.44$ | $1.2 + 22.45$ |
| $d = 6$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 0.19 | 0.19 | 0.19 |
| | $\frac{1}{64}$ | 0.32 | 0.32 | 0.33 |
| | $\frac{1}{128}$ | 1.35 | 1.33 | 1.31 |
| *AMG* | $\frac{1}{32}$ | $0.19 + 0.49$ | $0.19 + 0.49$ | $0.19 + 0.49$ |
| | $\frac{1}{64}$ | $0.32 + 2.63$ | $0.32 + 2.52$ | $0.33 + 2.53$ |
| | $\frac{1}{128}$ | $1.32 + 22.56$ | $1.33 + 21.98$ | $1.34 + 22.1$ |

Table E.2: Problem and $\mathcal{P}_{Schur}$ set-up times (sec.) - Test Problem 2

| | $\sigma$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.3 | 1.36 | 0.18 | 0.18 |
| *UMFPACK* | 0.5 | 0.18 | 0.18 | 0.2 |
| | 0.7 | 0.18 | 0.18 | 0.18 |
| | 0.3 | $1.46 + 0.56$ | $0.18 + 0.49$ | $0.19 + 0.49$ |
| *AMG* | 0.5 | $0.18 + 0.49$ | $0.18 + 0.48$ | $0.18 + 0.49$ |
| | 0.7 | $0.18 + 0.49$ | $0.18 + 0.5$ | $0.19 + 0.49$ |
| $d = 6$ | | | | |
| | 0.3 | 0.19 | 0.19 | 0.2 |
| *UMFPACK* | 0.5 | 0.19 | 0.19 | 0.2 |
| | 0.7 | 0.19 | 0.2 | 0.2 |
| | 0.3 | $0.2 + 0.49$ | $0.19 + 0.49$ | $0.21 + 0.49$ |
| *AMG* | 0.5 | $0.19 + 0.49$ | $0.19 + 0.49$ | $0.21 + 0.49$ |
| | 0.7 | $0.19 + 0.5$ | $0.19 + 0.49$ | $0.21 + 0.49$ |

Table E.3: Problem and $\mathcal{P}_{Schur}$ set-up times (sec.) - Test Problem 3

| | $\delta = \frac{\sigma}{\mu}$ | $p = 2$ | $p = 3$ | $p = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| | 0.3 | 0.2 | 0.17 | 0.17 |
| *UMFPACK* | 0.5 | 0.17 | 0.17 | 0.17 |
| | 0.7 | 0.17 | 0.17 | 0.17 |
| | 0.7,0.5,0.6,0.7 | 0.17 | 0.17 | 0.17 |
| | 0.5 | $1.47 + 0.71$ | $0.17 + 0.58$ | $0.17 + 0.57$ |
| *AMG* | 0.7 | $0.17 + 0.57$ | $0.17 + 0.57$ | $0.17 + 0.58$ |
| | 1.0 | $0.17 + 0.58$ | $0.17 + 0.58$ | $0.17 + 0.58$ |
| | 1.0,0.7,0.5,1.0 | $0.17 + 0.58$ | $0.17 + 0.57$ | $0.17 + 0.58$ |
| $d = 6$ | | | | |
| | 0.3 | 0.19 | 0.18 | 0.19 |
| *UMFPACK* | 0.5 | 0.18 | 0.19 | 0.19 |
| | 0.7 | 0.18 | 0.18 | 0.2 |
| | 0.7,0.5,0.6,0.7 | 0.18 | 0.19 | 0.2 |
| | 0.5 | $0.19 + 0.57$ | $0.18 + 0.58$ | $0.19 + 0.58$ |
| *AMG* | 0.7 | $0.18 + 0.57$ | $0.19 + 0.58$ | $0.2 + 0.57$ |
| | 1.0 | $0.18 + 0.57$ | $0.18 + 0.57$ | $0.2 + 0.58$ |
| | 1.0,0.7,0.5,1.0 | $0.18 + 0.57$ | $0.19 + 0.57$ | $0.2 + 0.58$ |

## E.2 Set-up times for test problems and $\mathcal{P}_{Schur}$ preconditioner - Non-linear case

Table E.4: Problem and $\mathcal{P}_{Schur}$ set-up times (sec.) - Test Problem 1

| | $h$ | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 0.78 | 1.59 | 5.68 |
| | $\frac{1}{64}$ | 1.32 | 3.9 | 11.32 |
| | $\frac{1}{128}$ | 5.7 | 15.69 | 38.74 |
| *AMG* | $\frac{1}{32}$ | $0.79 + 0.6$ | $1.58 + 0.49$ | $5.86 + 0.56$ |
| | $\frac{1}{64}$ | $1.31 + 2.65$ | $3.92 + 2.5$ | $11.17 + 2.51$ |
| | $\frac{1}{128}$ | $5.63 + 21.37$ | $15.86 + 22.7$ | $39.06 + 21.78$ |
| $d = 6$ | | | | |
| *UMFPACK* | $\frac{1}{32}$ | 1.95 | 22.6 | 210.17 |
| | $\frac{1}{64}$ | 4.25 | 33.19 | 254.85 |
| | $\frac{1}{128}$ | 16.11 | 82.01 | 431.86 |
| *AMG* | $\frac{1}{32}$ | $1.98 + 0.5$ | $22.31 + 0.49$ | $213.07 + 0.51$ |
| | $\frac{1}{64}$ | $4.14 + 2.5$ | $32.89 + 2.53$ | $248.62 + 2.64$ |
| | $\frac{1}{128}$ | $15.5 + 24.85$ | $83.42 + 22.19$ | $595.74 + 35.51$ |

Table E.5: Problem and $\mathcal{P}_{Schur}$ set-up times (sec.) - Test Problem 2

| | $\sigma$ | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ | | | | |
| *UMFPACK* | 0.3 | 1.15 | 1.67 | 5.82 |
| | 0.5 | 0.52 | 1.65 | 5.75 |
| | 0.7 | 0.53 | 1.64 | 5.73 |
| | 0.9 | 0.53 | 1.65 | 5.66 |
| *AMG* | 0.3 | $1.56 + 0.6$ | $1.65 + 0.51$ | $5.81 + 0.56$ |
| | 0.5 | $0.52 + 0.51$ | $1.66 + 0.51$ | $5.72 + 0.53$ |
| | 0.7 | $0.52 + 0.5$ | $1.63 + 0.51$ | $5.81 + 0.51$ |
| | 0.9 | $0.52 + 0.51$ | $1.64 + 0.51$ | $5.75 + 0.52$ |
| $d = 6$ | | | | |
| *UMFPACK* | 0.3 | 2.05 | 22.57 | 211.48 |
| | 0.5 | 1.95 | 22.81 | 212.87 |
| | 0.7 | 1.97 | 23.37 | 210.49 |
| | 0.9 | 1.99 | 23 | 216.49 |
| *AMG* | 0.3 | $1.99 + 0.51$ | $22.8 + 0.52$ | $209 + 0.51$ |
| | 0.5 | $1.94 + 0.51$ | $23.11 + 0.5$ | $214.48 + 0.62$ |
| | 0.7 | $1.95 + 0.52$ | $22.68 + 0.51$ | $213.19 + 0.52$ |
| | 0.9 | $1.96 + 0.52$ | $22.88 + 0.51$ | $212.63 + 0.51$ |

Table E.6: Problem and $\mathcal{P}_{Schur}$ set-up times (sec.) - Test Problem 3

|  | $\delta$ | $p_u = 2$ | $p_u = 3$ | $p_u = 4$ |
|---|---|---|---|---|
| $d = 4$ |  |  |  |  |
| *UMFPACK* | 0.5 | 0.63 | 2.18 | 6.38 |
|  | 0.7 | 0.6 | 1.84 | 6.25 |
|  | 1.0 | 0.58 | 1.86 | 6.32 |
|  | 1.0,0.7,0.5,1.0 | 0.58 | 1.88 | 6.29 |
| *AMG* | 0.5 | $0.73 + 0.53$ | $1.86 + 0.5$ | $6.33 + 0.51$ |
|  | 0.7 | $0.57 + 0.51$ | $1.9 + 0.51$ | $6.42 + 0.51$ |
|  | 1.0 | $0.57 + 0.6$ | $1.84 + 0.51$ | $6.25 + 0.55$ |
|  | 1.0,0.7,0.5,1.0 | $0.57 + 0.51$ | $1.86 + 0.5$ | $6.32 + 0.53$ |
| $d = 6$ |  |  |  |  |
| *UMFPACK* | 0.5 | 2.26 | 24.13 | 221.01 |
|  | 0.7 | 2.19 | 23.92 | 215.01 |
|  | 1.0 | 2.2 | 23.57 | 217.12 |
|  | 1.0,0.7,0.5,1.0 | 2.18 | 23.39 | 221.24 |
| *AMG* | 0.5 | $2.27 + 0.52$ | $24.06 + 0.51$ | $222.52 + 0.52$ |
|  | 0.7 | $2.19 + 0.52$ | $23.86 + 0.52$ | $216.1 + 0.53$ |
|  | 1.0 | $2.19 + 0.52$ | $23.78 + 0.53$ | $221.73 + 0.57$ |
|  | 1.0,0.7,0.5,1.0 | $2.19 + 0.52$ | $23.99 + 0.53$ | $220.69 + 0.55$ |

# Appendix F

# Notation

## F.1 Notation for Chapter 2

| | |
|---|---|
| $u(\mathbf{x})$ | $:=$ potential head |
| $D$ | $:=$ physical domain |
| $\Gamma$ | $:=$ boundary of physical domain $D$ |
| $\Gamma_D$ | $:=$ Dirichlet boundary of $\Gamma$ |
| $\Gamma_N$ | $:=$ Neumann boundary of $\Gamma$ |
| $\mathcal{C}$ | $:=$ hydraulic conductivity tensor |
| $\mathbf{n}$ | $:=$ unit outward normal vector to $\Gamma_N$ |
| $g(\mathbf{x})$ | $:=$ prescribed constant head on $\Gamma_D$ |
| $\mathbf{q}$ | $:=$ fluid discharge (flux) |
| $q_x$ | $:=$ $x$-component of fluid discharge |
| $q_y$ | $:=$ $y$-component of fluid discharge |
| $L^2(D)$ | $:= \{w : w \text{ is defined on } D \text{ and } \int_D w^2 dD < \infty\}$ |
| $L^2(D)^d$ | $:= \{\mathbf{v} : v_i \in L^2(D),\ i = 1, \ldots, d\}$ |

$$
\begin{aligned}
H^1(D) &:= \{w : w \in L^2(D) \text{ and } \tfrac{\partial w}{\partial x_i} \in L^2(D), i = 1, \ldots, d\} \\
H^1_0(D) &:= \{w \in H^1(D) : w = 0 \text{ on } \Gamma\} \\
H^1_{0,D}(D) &:= \{w \in H^1(D) : w = 0 \text{ on } \Gamma_D\} \\
\mathbf{v} &:= (v_1, \ldots, v_d)^T \\
H(div; D) &:= \{\mathbf{v} : \mathbf{v} \in L^2(D)^d \text{ and } \nabla \cdot \mathbf{v} \in L^2(D)\} \\
H^{\frac{1}{2}}(\Gamma) &:= \{g : g = w_\Gamma \text{ for some } w \in H^1(D)\} \\
H^{-\frac{1}{2}}(\Gamma) &:= \{q : q = (\mathbf{v} \cdot \mathbf{n})_\Gamma \text{ for some } \mathbf{v} \in H(div; D)\} \\
H_{0,N}(div; D) &:= \{\mathbf{v} \in H(div; D) : \langle \mathbf{v} \cdot \mathbf{n}, w \rangle = 0, \ \forall w \in H^1_{0,D}(D)\} \\
T^h &:= \text{Partition of } D \\
K &:= \text{Finite element of } T^h \\
h &:= \text{Discretisation parameter} \\
E^h &:= \text{collection of numbered edges } (\mathcal{D} = 2) \text{ or faces } (\mathcal{D} = 3) \\
\mathcal{I}^h \subset E^h &:= \{e \in E^h : e \not\subset \Gamma_D\} \\
RT^0(K) &:= \left\{\mathbf{v} : \mathbf{v}(\mathbf{x}) = \tfrac{\mathbf{B}\hat{\mathbf{v}}(\boldsymbol{\xi})}{J} \ \forall \ \boldsymbol{\xi} \in \hat{K} \text{ and } \hat{\mathbf{v}} \in RT^0(\hat{K})\right\} \\
RT^0(D; T^h) &:= \{\mathbf{v} \in H(div; D) : \mathbf{v}|_K \in RT^0(K) \ \forall K \in T^h\} \\
\mathcal{M}^0 &:= \left\{\mathbf{v} \in L^2(D)^d \text{ and } \mathbf{q}|_K \in RT^0(K) \ \forall K \in T^h\right\} \\
V^h &:= \mathcal{M}^0 \cap H_{0,N}(div; D) = \left\{\mathbf{v} \in RT^0(D; T^h) \text{ and } \mathbf{v} \cdot \mathbf{n}|_{\Gamma_N} = 0\right\} \\
W^h &:= \{w \in L^2(D) : w|_K \in M^0(K) \ \forall \ K \in T^h\} \\
\phi_j, i = 1, \ldots, n &:= \text{Scalar basis functions for } W^h \\
\boldsymbol{\varphi}_i, i = 1, \ldots, m &:= \text{Vector basis functions for } V^h \\
A_{i,j} &:= \text{global weighted velocity matrix} \\
B_{k,i} &:= \text{divergence operator matrix} \\
\Lambda_0\left(E^h\right) &:= \left\{\lambda^h : \lambda^h|_e \in \Lambda_0(e) \forall e \in E^h\right\} \\
\Lambda_{0,\Gamma_D} &:= \{\lambda \in \Lambda\left(E_h\right) : \lambda = 0 \text{ on } \Gamma_D\}, \\
\Lambda_{g,\Gamma_D} &:= \left\{\lambda \in \Lambda\left(E_h\right) : \lambda = g^h \text{ on } \Gamma_D\right\} \\
\mu_j, i = 1, \ldots, l &:= \text{Scalar basis functions for } \Lambda_{0,\Gamma_D}
\end{aligned}
$$

## F.2 Notation for Chapter 4 and 5

| | |
|---|---|
| $\Omega$ | $:=$ set of random events |
| $\Im$ | $:=$ minimal $\sigma$-algebra |
| $Pr$ | $:=$ probability measure |
| $(\Omega, \Im, Pr)$ | $:=$ probability space |
| $u(\mathbf{x}, \omega)$ | $:=$ random potential solution |
| $\mathbf{q}(\mathbf{x}, \omega)$ | $:=$ random flux solution |
| $\mathcal{C}(\mathbf{x}, \omega)$ | $:=$ random conductivity coefficient |
| $\mu(\mathbf{x})$ | $:=$ mean conductivity value |
| $\sigma$ | $:=$ standard deviation of conductivity random field |
| $\beta_i(\mathbf{x}), \lambda_i$ | $:=$ eigenfunctions and eigenvalues of the covariance function |
| $\xi_i$ | $:=$ normal or uniform random variables |
| $\rho(\mathbf{x}, \mathbf{x}')$ | $:=$ correlation function of $\mathcal{C}(\mathbf{x}, \cdot)$ |
| $L^2(\Omega)$ | $:= \{w : \ w \text{ is defined on } \Omega \text{ and } \int_\Omega w^2 d\Omega < \infty\}$ |
| $W$ | $:= H_0^1(D) \otimes L^2(\Omega)$ |
| $S^h$ | $\subset H_0^1(D)$ |
| $T^h$ | $\subset L^2(\Omega)$ |
| $W^h$ | $:= S^h \otimes T^h \subset W = H_0^1(D) \otimes L^2(\Omega)$ |
| $V$ | $:= \{\mathbf{v}(\mathbf{x}, \boldsymbol{\xi}) \in H(div; D) \otimes L^2(\Omega) : \mathbf{v}(\mathbf{x}, \boldsymbol{\xi}) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N \times \Omega\}$ |
| $Y^h$ | $\subset H(div; D)$ |
| $V^h$ | $:= Y^h \otimes T^h \subset V = H(div; D) \otimes L^2(\Omega)$ |
| $X^h$ | $:= L^2(D)$ |
| $W^h$ | $:= X^h \otimes T^h \subset W = L^2(D) \otimes L^2(\Omega)$ |
| $Z^h$ | $:=$ partition of $D$ |

$\triangle$ := Finite element of $Z^h$

$N_u$ := number of nodes in $Z^h$

$N_e$ := number of elements in $Z^h$

$N_{edg}$ := number of edges in $Z^h$

$p$ := order of complete polynomials

$P := \frac{(d+p)!}{d!p!}$

$L_i(\xi_j)$ := univariate Legendre polynomials

$H_i(\xi_j)$ := univariate Hermite polynomials

$\{\chi_i\}, i = 1, \ldots, P$ := stochastic basis functions

$\boldsymbol{\alpha}_{(i,j)}$ := multi-index

$(K_0)_{r,s} = \int_D \mu \nabla \phi_r(\mathbf{x}) \nabla \phi_s(\mathbf{x}) d\mathbf{x}$ := FEM mean stiffness matrix

$A^*_{i,i} := \langle \chi_i \rangle^2 \otimes K_0, \; i = 1, \ldots, P$

$(K_l)_{r,s} = \sigma \sqrt{\lambda_l} \int_D \beta_l(\mathbf{x}) \nabla \phi_r(\mathbf{x}) \nabla \phi_s(\mathbf{x}) d\mathbf{x}$ := FEM 'fluctuation' stiffness matrices

$A^\star_{i,j} := \sum_{l=1}^{d} [\langle \xi_l \chi_i \chi_j \rangle] \otimes K_l$

$A := G_0 \otimes K_0 + \sum_{k=1}^{d} G_k \otimes K_k$

$\mathbf{z}^0$ := initial guess

$\mathbf{r}$ := residual vector

$(K_0)_{r,s} = \int_D \frac{1}{\mu} \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}$ := MFEM mean stiffness matrix

$A^*_{i,i} := \langle \chi_i \rangle^2 \otimes K_0$

$(K_l)_{r,s} = \sigma \sqrt{\lambda_l} \int_D \beta_l(\mathbf{x}) \psi_r(\mathbf{x}) \psi_s(\mathbf{x}) d\mathbf{x}$ := MFEM 'fluctuation' stiffness matrices

$(B_0)_{r,s} = \int_D \phi_r(\mathbf{x}) \nabla \cdot \psi_s(\mathbf{x}) d(\mathbf{x})$ := divergence operator

$B_{i,i} := \langle \chi_i \rangle^2 \otimes B_0$

$\mathcal{L}(\mathbf{x}, \omega)$ := random conductivity coefficient (Lognormal)

$p_u$ := order of complete polynomials for $u$

$p_{\mathcal{L}}$ := order of complete polynomials for $\mathcal{L}$